1976

On the stability of plane Poiseuille flow.

Newby, Lewis Raymond

ON THE STABILITY OF PLANE POISEUILLE FLOW

Lewis Raymond Newby

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ON THE STABILITY
OF
PLANE POISEUILLE FLOW

by

Lewis Raymond Newby

March 1976

Thesis Advisor: T. H. Gawain

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	
2. GOVT ACCESSION NO.	
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	On the Stability of Plane Poiseuille Flow
5. TYPE OF REPORT & PERIOD COVERED	Master's Thesis; March 1976
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	Lewis Raymond Newby
8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Naval Postgraduate School Monterey, California 93940
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	Naval Postgraduate School Monterey, California 93940
12. REPORT DATE	March 1976
13. NUMBER OF PAGES	65
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	Naval Postgraduate School Monterey, California 93940
15. SECURITY CLASS. (of this report)	Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)	Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)	Stability Plane Poiseuille Incipient Instability Critical Instability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The stability of plane Poiseuille flow was studied using theory developed by Harrison. A similarity transformation was introduced which reduces computation time and provides better insight into the basic relations. The stability of the flow was examined from a Lagrangian viewpoint. Instability was found to be progressive in nature and three distinct levels

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

were identified, namely incipient, critical, and fully developed instability.

Results show that the critical Reynolds number can be lowered indefinitely if certain types of perturbations occur. Specifically these involve relatively abrupt changes in amplitude. This provides a possible explanation for the disagreement between earlier theory and experiment.

On the Stability
of
Plane Poiseuille Flow

by

Lewis Raymond Newby
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1976

ABSTRACT

The stability of plane Poiseuille flow was studied using theory developed by Harrison. A similarity transformation was introduced which reduces computation time and provides better insight into the basic relations. The stability of the flow was examined from a Lagrangian viewpoint. Instability was found to be progressive in nature and three distinct levels were identified, namely incipient, critical, and fully developed instability.

Results show that the critical Reynolds number can be lowered indefinitely if certain types of perturbations occur. Specifically these involve relatively abrupt changes in amplitude. This provides a possible explanation for the disagreement between earlier theory and experiment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

All quantities are expressed in dimensionless form by the use of a natural system of consistent units in which channel semi-height is the unit of length, the volumetric mean velocity of the fluid is the unit of velocity, and the density of the fluid is the unit of density. Then all other consistent derived units are fixed accordingly.

A^*	wave number amplitude defined in Eq. 2.4
e	2.71828...base of the natural logarithms.
G	stability parameter defined in Eq. 1-30.
G, H	complex stream functions.
G^*, H^*	transformed stream functions.
$I(y)$	complex auxiliary function defined in Eq. 2-14.
i	$(-1)^{1/2}$ the imaginary unit.
$\bar{i}, \bar{j}, \bar{k}$	unit vectors along x, y, and z axes, respectively.
$J(y)$	complex auxiliary function defined in Eq. 2-15.
$J^*(y)$	transformed auxiliary function defined in Eq. 2-27.
Re	Reynolds number based on volumetric mean velocity and channel semi-height.
Re*	transformed Reynolds number defined in Eq. 2-18.
$T(y)$	complex auxiliary function defined in Eq. 2-13.
t	time.
U	flow velocity.

$\overline{W}$	complex vector potential of perturbation flow defined in Eq. 3-2.
x,y,z	coordinates in direction of mean flow, normal to walls and transverse to the mean flow, respectively.
x',y,z	coordinates in moving reference frame.
α	complex wave number of the perturbation in x direction.
α^*	transformed wave number defined in Eq. 2-3.
β	complex wave number of the perturbation in the z direction.
γ	complex frequency of the perturbation in a fixed reference frame.
γ'	complex frequency of the perturbation in the moving reference frame.
γ^*	transformed complex frequency defined in Eq. 2-21.
θ	phase angle parameter defined in Eq. 2-12.
κ	amplitude parameter defined in Eq. 2-19,
Λ	angle of plane of perturbation with respect to xy plane.
Λ_R	angle of resultant growth wave number vector $\overline{\lambda}_R$ with respect to x axis.
Λ_I	angle of oscillation wave number vector $\overline{\lambda}_I$ with respect to x axis.
$\overline{\lambda}_R$	growth wave number vector.
$\overline{\lambda}_I$	oscillation wave number vector.
ϕ	wave number phase angle defined in Eq. 2-9.
ϕ^*	wave number phase angle defined in Eq. 2-7.
ψ	wave number phase angle defined in Eq. 2-2.

I. THEORETICAL BACKGROUND AND APPROACH

A. BACKGROUND

This research deals with the instability of plane Poiseuille flow, that is, plane flow between infinite parallel plates. The mean velocity of this flow is given by the expression

$$U = \frac{3}{2}(1-y^2) \ . \tag{1-1}$$

The stability of such a flow field is determined by superimposing upon it an appropriate perturbation and determining whether this perturbation tends to grow or decay over time. In the present case the perturbations are expressed by a complex vector potential which is taken to be of the form

$$\overline{W} = [\overline{j}G(y) + \overline{k}H(y)] \exp(\alpha x + \beta z + \gamma t) \ . \tag{1-2}$$

The complex constants α and β fix the spatial characteristics of the perturbation and may be arbitrarily prescribed whereas the complex constant γ fixes the response in time and must be found by solving the vorticity transport equation. Moreover, since α, β and γ are all complex, they can be resolved into real and imaginary components in the form

$$\alpha = \alpha_R + i\alpha_I \tag{1-3}$$

$$\beta = \beta_R + i\beta_I \tag{1-4}$$

$$\gamma = \gamma_R + i\gamma_I \ . \tag{1-5}$$

Thus the spatial characteristics of the perturbation are seen to be completely defined by the four constants α_R, α_I, β_R, β_I . In addition, the mean flow is characterized by its Reynolds number Re.

Harrison's original analysis [Ref. 1] showed that the perturbation growth rate in time as seen by a fixed observer, and as expressed by the parameter γ_R, is a definite function of the five parameters α_R, α_I, β_R, β_I and Re, which characterize the perturbations and the flow. Thus

$$\gamma_R = \gamma_R[\alpha_R, \alpha_I, \beta_R, \beta_I, Re] \ . \tag{1-6}$$

B. PROGRESSIVE INSTABILITY

In a further development of Harrison's original approach, Section II of this thesis shows that three significant levels of instability can be defined which are termed incipient, critical and fully developed instability. The definition of these terms depends, in part, on the algebraic sign of α_R. However, Harrison showed that negative values of α_R have a definitely destabilizing effect. Consequently the present

analysis is restricted to the critical case of negative α_R. For this case the three levels of instability correspond to the following levels of γ_R, namely

$$\text{Incipient Instability} \quad \gamma_R)_I = 0 \tag{1-7}$$

$$\text{Critical Instability} \quad \gamma_R)_C = -\alpha_R \tag{1-8}$$

$$\text{Fully Developed Instability} \quad \gamma_R)_D = -\frac{3}{2}\alpha_R \,. \tag{1-9}$$

Numerical solution of the vorticity transport equations enables us to find the three corresponding Reynolds numbers at which the above stability levels are reached. Thus

$$\text{Incipient Instability} \quad Re)_I = RE_I[\alpha_R, \alpha_I, \beta_R, \beta_I] \tag{1-10}$$

$$\text{Critical Instability} \quad Re)_C = Re_C[\alpha_R, \alpha_I, \beta_R, \beta_I] \tag{1-11}$$

$$\text{Fully Developed Instability} \quad Re)_D = Re_D[\alpha_R, \alpha_I, \beta_R, \beta_I] \tag{1-12}$$

C. TRANSFORMATION OF PARAMETERS

In Section II of this thesis, a transformation is developed which relates the original parameters α_R, α_I, β_R, β_I, Re and γ_R to an alternative set of parameters which are symbolized as A^*, $\emptyset^*$, θ, Re^* and γ_R^*. This transformation

can be expressed in several alternative but equivalent forms. In the present context it is convenient to write

$$\alpha_R = A^* \cos(\phi^* + \theta) \tag{1-13}$$

$$\alpha_I = A^* \sin(\phi^* + \theta) \tag{1-14}$$

$$\beta_R^2 = \frac{A^{*2}}{2} \{[(1-\kappa^2)^2 + 4\kappa^2 \sin^2\theta]^{1/2} + \cos 2\phi^* - \kappa^2 \cos 2(\phi^* + \theta)\} \tag{1-15}$$

$$\beta_I^2 = \frac{A^{*2}}{2} \{[(1-\kappa^2)^2 + 4\kappa^2 \sin^2\theta]^{1/2} - \cos 2\phi^* + \kappa^2 \cos 2(\phi^* + \theta)\} \tag{1-16}$$

$$Re = Re^*/\kappa \tag{1-17}$$

and

$$\gamma_R = \kappa \gamma_R^* . \tag{1-18}$$

The important fact about this new set of parameters, which are somewhat loosely termed the starred parameters, is that their use permits the fundamental vorticity transport equation to be simplified. Specifically, the relation analogous to Eq. 1-1 reduces to

$$\gamma_R^* = \gamma_R^*[A^*, \phi^*, \theta, Re^*] \tag{1-19}$$

The relations analogous to Eqs. 1-7, 1-8, and 1-9 reduce to

Incipient Instability $\gamma_R^*)_I = 0$ (1-20)

Critical Instability $\gamma_R^*)_C = -\alpha_R^* = -\frac{3}{2} A^* \cos(\phi^* + \theta)$ (1-21)

Fully Developed Instability $\gamma_R^*)_D = -\frac{3}{2} \alpha_R^* = -\frac{3}{2} A^* \cos(\phi^* + \theta)$ (1-22)

Finally, the relations analogous to Eqs. 1-5, 1-6, and 1-7 simplify to

Incipient Instability $Re_I^* = RE_I^*[A^*, \phi^*, \theta]$ (1-23)

Critical Instability $Re_C^* = Re_C^*[A^*, \phi^*, \theta]$ (1-24)

Fully Developed Instability $Re_D^* = Re_D^*[A^*, \phi^*, \theta]$ (1-25)

The remarkable feature of Eqs. 1-19 through 1-25 is that none of these relations involve the parameter κ. Thus the number of independent parameters has been reduced from four in Eqs. 1-10, 1-11, and 1-12 to three in Eqs. 1-23, 1-24, and 1-25. This represents a very significant simplification of the problem, especially in view of the tremendous computational burden which these equations involve.

D. PHYSICAL SIGNIFICANCE OF STABILITY BOUNDARIES

The stability boundaries $Re^*)_I$, $Re^*)_C$, and $Re^*)_D$ symbolized by Eqs. 1-23, 1-24, and 1-25 have physical significance which can be interpreted in a straightforward manner. Eq. 1-17 shows that Re^* can be regarded as the value of Re which corresponds to the reference case $\kappa = 1$. Thus $Re^*)_I$, for example, is the Reynolds number of incipient instability for a perturbation which is characterized by the given values of parameters A^*, $\emptyset^*$, and θ and by the reference value $\kappa = 1$. Since the transformed quantities A^*, $\emptyset^*$, and θ may, at first, seem to be somewhat abstract in character, it is helpful to go back to Eqs. 1-13, 1-14, 1-15, and 1-16 and ascertain the corresponding values of the original untransformed parameters α_R, α_I, β_R, β_I.

However, there is far more to this solution than just the above reference case, $\kappa = 1$. In this connection, Eq. 1-17, when taken in conjunction with Eqs. 1-23, 1-24, and 1-25, reveals a most important result. It shows that for given values of parameters A^*, $\emptyset^*$, and θ, and hence for the corresponding values of $Re^*)_I$, $Re^*)_C$, or $Re^*)_D$, the corresponding actual Reynolds numbers $Re)_I$, $Re)_C$, or $Re)_D$ can be lowered indefinitely, simply by increasing parameter κ to any desired extent. Notice that such a shift of the stability boundaries, while it involves no change in parameters A^*, $\emptyset^*$, and θ, does involve changes

in α_R, α_I, β_R, and β_I. It is therefore important to summarize the nature of these changes in the clearest possible way.

For this purpose, it is useful to regroup the four quantities α_R, α_I, β_R, and β_I into two vectors $\bar{\lambda}_R$ and $\bar{\lambda}_I$ defined as follows:

$$\bar{\lambda}_R = \bar{i}\alpha_R + \bar{k}\beta_R \tag{1-26}$$

$$\bar{\lambda}_I = \bar{i}\alpha_I + \bar{k}\beta_I \tag{1-27}$$

($\bar{i}$ and $\bar{k}$ are unit vectors in the x and z directions, respectively.)

Clearly $\bar{\lambda}_R$ represents the spatial growth rate in vector form, that is, in terms of magnitude and direction, while $\bar{\lambda}_I$ represents the spatial oscillation rate in like terms. Each of these vectors is characterized by a magnitude and a direction. In this case the magnitudes λ_R and λ_I turn out to be governed by the relations

$$\lambda_R^2 = \frac{A^{*2}}{2}\{([(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}-(1-\kappa^2))+2\cos^2\emptyset^*\} \tag{1-28}$$

$$\lambda_I^2 = \frac{A^{*2}}{2}\{([(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}-(1-\kappa^2))+2\sin^2\emptyset^*\} . \tag{1-29}$$

Likewise the two corresponding angles Λ_R and Λ_I which the above vectors make with respect to the x axis turn out to

be governed by the relations

$$\tan^2\Lambda_R = \frac{[(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}+\cos2\emptyset^*-\kappa^2\cos2(\emptyset^*+\ )}{\kappa^2[1+\cos2(\emptyset^*+\theta)]} \tag{1-30}$$

$$\tan^2\Lambda_I = \frac{[(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}-\cos2\emptyset^*+\kappa^2\cos2(\emptyset^*+\theta)}{\kappa^2[1-\cos2(\emptyset^*+\theta)]} \tag{1-31}$$

Thus the spatial form of the perturbations is now fully characterized by the four transformed parameters λ_R, λ_I, Λ_R, and Λ_I which are in some respects more convenient than the four original parameters α_R, α_I, β_R, and β_I.

E. EFFECT OF VARYING PARAMETERS

The connection between the above perturbation characteristics and the Reynolds number is still expressed by the relation

$$Re = \frac{Re^*}{\kappa} . \tag{1-32}$$

It is very instructive to study the trends revealed by Eqs. 1-28 through 1-32 when parameters A^*, $\emptyset^*$, and θ are held constant while κ is allowed to increase. Eq. 1-32 reveals that $Re)_I$, $Re)_C$, and $Re)_D$ can be decreased indefinitely in this manner. On the other hand, Eq. 1-28 reveals that any such decrease in Reynolds number always entails a corresponding increase in the quantity λ_R. Recall that λ_R

represents exponential growth rate in space. This says then that stability boundaries are not absolute in character but depend significantly on the "abruptness" of the perturbation in space as measured by parameter λ_R. The greater this abruptness parameter, the lower the Reynolds number at which instability can occur.

Conversely, if the permissible magnitude of λ_R be limited in some definite manner, the reduction that can be achieved in $Re)_I$, $Re)_C$ and $Re)_D$ will be correspondingly limited as well. In that case, a systematic exploration over appropriate ranges of the parameters A^*, ϕ^*, and θ should ultimately reveal corresponding ultimate stability limits and, in particular, some critical Reynolds number below which no instabilities occur. Notice, however, that such a critical Reynolds number is never absolute, but is always contingent upon the restriction that has been placed upon parameter λ_R.

1. Restrictions on λ_R

The most obvious and direct restriction that can be placed on λ_R is simply to limit it to some fixed value or, for study and comparison purposes, to some series of successive fixed values. In general, the boundaries which correspond to incipient, critical and fully developed instability will then depend on the designated value of λ_R. The higher this value, the lower the values of Re at which the above boundaries will occur.

Any such restriction of the magnitude of λ_R implies a corresponding restriction on κ. To show this, invert Eq. 1-28, solving for κ as a function of λ_R. The result is

$$\kappa^2 = \frac{\left[\left(\frac{\lambda R}{A^*}\right)^2 - \cos^2\phi^*\right]\left[\left(\frac{\lambda R}{A^*}\right)^2 + \sin^2\phi^*\right]}{\left[\left(\frac{\lambda R}{A^*}\right)^2 - \cos^2\phi^* + \sin^2\phi^*\right]} \tag{1-33}$$

This relation may be used in connection with Eq. 1-32 to express the final Reynolds number at which the designated stability boundary is reached. For incipient instability, for example, this boundary may be expressed in the form

$$Re)_I = \sqrt{\frac{\left[\left(\frac{\lambda_R}{A^*}\right)^2 - \cos^2\phi^* + \sin^2\theta\right]}{\left[\left(\frac{\lambda_R}{A^*}\right)^2 - \cos^2\phi^*\right]\left[\left(\frac{\lambda_R}{A^*}\right)^2 + \sin^2\phi^*\right]}}\; Re_I^*(A^*,\phi^*,\theta) \tag{1-34}$$

Analogous expressions apply to the boundaries for critical and fully developed instability.

Equation 1-34 shows quite clearly that the stability condition in question depends on the three characteristic parameters A^*, ϕ^*, θ of the perturbation as well as on the limiting value assigned to parameter λ_R. This procedure

of calculating stability boundaries for various assumed combinations of A^*, $\emptyset^*$, θ and λ_R has been carried out for several typical cases and the detailed results are summarized in Section IV of this paper. Of course, these examples, while representative, merely scratch the surface of the stability problem. The complication remains that the true and ultimate stability boundary represents the lowest possible Reynolds number at which an instability can just occur. This implies that all possible combinations of parameters A^*, $\emptyset^*$ and θ must be examined to determine the particular combination which, for a given limit on λ_R, yields a stability boundary at the lowest possible value of Re. In other words, the true stability boundary amounts to the envelope of all the individual stability boundaries.

Each individual boundary is characterized by some specified combination of A^*, $\emptyset^*$ and θ and, of course, also of λ_R. Since there is an unlimited number of such combinations, the amount of calculation involved in establishing the desired stability envelope is prodigious. Needless to say, no such attempt was made in the present thesis to accomplish anything so ambitious.

F. SCOPE OF PRESENT RESEARCH

The present research was restricted to the more modest and realistic aim of calculating stability boundaries for a few specific and typical combinations of A^*, $\emptyset^*$, θ and λ_R. This goal has been successfully attained.

G. PARAMETER G

1. Definition of Parameter

A detailed study of the relations summarized by Eqs. 1-28 through 1-32 reveals the possibility of expressing a restriction on the permissible magnitude of λ_R in a rather subtle and indirect way, through a change of variable. The particular algebraic form which the above relations assume suggests the utility of defining a new parameter, called G, as follows:

$$G^2 = 1/2 \{[(1-\kappa^2)^2 + 4\kappa^2\sin^2\theta]^{1/2} - (1-\kappa^2)\} \qquad (1\text{-}35)$$

This relation can be readily inverted to give

$$\kappa^2 = \frac{G^2(G^2 + 1)}{G^2 + \sin^2\theta} \qquad (1\text{-}36)$$

2. Utilization of Parameter G

Equations 1-35 and 1-36 may be used to eliminate parameter κ from Eqs. 1-28 through 1-32, replacing it by the new parameter G. In this way the following results are obtained.

The vector amplitudes λ_R and λ_I turn out to be related to the new parameter G in a fairly simple fashion.

Thus

$$\lambda_R = A^*[(G^2 + \cos^2\phi^*)]^{1/2} \tag{1-37}$$

$$\lambda_I = A^*[(G^2 + \sin^2\phi^*)]^{1/2} \tag{1-38}$$

On the other hand, the angles Λ_R and Λ_I are not simplified by the use of parameter G. Fortunately these quantities are less significant than the preceding ones. The governing equations become

$$\tan^2\Lambda_R = \frac{(G^2+\sin^2\theta)(G^2-\sin^2\phi^*)+G^2(G^2+1)\sin^2(\phi^*+\theta)}{G^2(G^2+1)\cos^2(\phi^*+\theta)} \tag{1-39}$$

and

$$\tan^2\Lambda_I = \frac{(G^2+\sin^2\theta)(G^2-\cos^2\phi^*)+G^2(G^2+1)\cos^2(\phi^*+\theta)}{G^2(G^2+1)\sin^2(\phi^*+\theta)} \tag{1-40}$$

The important Reynolds number relation below is once again simple. For definiteness it is written specifically for the case of incipient instability, by analogy with Eq. 1-34. Similar expressions apply also to the boundaries of critical and fully developed instability. Thus

$$Re)_I = \sqrt{\frac{G^2 + \sin^2\theta}{G^2(G^2 + 1)}}\, Re_I^*(A^*,\emptyset^*,\theta) \qquad (1\text{-}41)$$

Equation 1-41 shows that the stability depends on the particular parameters A^*, $\emptyset^*$, θ and on the limiting value assigned to G. Hence G plays a similar role in relation to Eq. 1-41 that λ_R plays in relation to Eq. 1-34.

The results summarized elsewhere in this thesis are presented primarily from the perspective expressed by Eq. 1-34. The alternative version shown by Eq. 1-41 is included in this discussion because of its theoretical interest, but this version is not used in the presentation of calculated results.

Notice that in either version, assuming some assigned limit for λ_R or G, an exploration is still required over the domain of parameters A^*, $\emptyset^*$ and θ to find the particular combination that yields the minimum Reynolds number. Of course, such extensive exploration could not be undertaken in the present thesis owing to time limitations.

H. REDUCTION TO CLASSICAL THEORY

It is pertinent to note that the classical theory of the stability of plane Poiseuille flow amounts to a special case of the more general theory discussed above. It

amounts, in fact, to the special case for which

$$\lambda_R = 0. \tag{1-42}$$

Study of Eq. 1-28 reveals that Eq. 1-42 can be satisfied if and only if we set

$$\theta = 0. \tag{1-43}$$

and

$$\emptyset^* = \frac{\pi}{2} . \tag{1-44}$$

From Eqs. 1-28, 1-32, and 1-42 we may infer also that

$$\kappa = 1 . \tag{1-45}$$

It then follows from Eq. 1-35 that

$$G = 0.$$

Moreover, we also find under these conditions that

$$\alpha_R = 0 \ , \ \beta_R = 0 \ , \text{ and } \beta_I = 0. \tag{1-46}$$

It is evident that the general theory discussed in this thesis is immensely more comprehensive than the classical theory as limited by Eqs. 1-42 through 1-46.

II. SIMILARITY TRANSFORMATION

The perturbation characteristics are fully defined, as in Ref. 1, by the four real parameters α_R, α_I, β_R, and β_I. α_R and α_I are the components of the complex wave number of the perturbation in the x direction and β_R and β_I are the components of the complex wave number of the perturbation in the z direction.

The above parameters satisfy the following relations:

$$\alpha = \kappa A^* e^{i\phi} = \alpha_R + i\alpha_I \tag{2-1}$$

$$\beta = \sigma A^* e^{i\psi} = \beta_R + i\beta_I \tag{2-2}$$

A very useful alternative set of parameters is α_R^*, α_I^*, θ and κ.

α^{*2} is defined by

$$\alpha^{*2} = \alpha^2 + \beta^2 . \tag{2-3}$$

A^*, ϕ^*, α_R^* and α_I^* are defined by

$$\alpha^* = A^* e^{i\phi^*} = \alpha_R^* + i\alpha_I^* . \tag{2-4}$$

α_R^* and α_I^* are the components of α^*, the transformed complex wave number parameter and κ is the transformed perturbation amplitude parameter.

κ and θ are defined by

$$\alpha = \alpha^* \kappa e^{i\theta}. \tag{2-5}$$

A^* and κ are positive by definition.

Given the original parameters α_R, α_I, β_R and β_I, the transformed parameters α_R^*, α_I^*, θ and κ may be deduced from equations 2-1 through 2-5 as follows:

$$A^* = [(\alpha_R^2 - \alpha_I^2 + \beta_R^2 - \beta_I^2)^2 + 4(\alpha_R\alpha_I + \beta_R\beta_I)^2]^{1/4} \tag{2-6}$$

$$\phi^* = 1/2 \arctan\left[\frac{2(\alpha_R\alpha_I + \beta_R\beta_I)}{\alpha_R^2 - \alpha_I^2 + \beta_R^2 - \beta_I^2}\right] \tag{2-7}$$

$$\kappa A^* = [\alpha_R^2 + \alpha_I^2]^{1/2} \tag{2-8}$$

$$\phi = \arctan\left(\frac{\alpha_I}{\alpha_R}\right) \tag{2-9}$$

$$\alpha_R^* = A^* \cos \phi^* \tag{2-10}$$

$$\alpha_I^* = A^* \sin \phi^* \qquad (2\text{-}11)$$

$$\theta = (\phi - \phi^*) \qquad (2\text{-}12)$$

The governing equations of Ref. 1, using the five independent parameters, Re, α_R, α_I, β_R, β_I are as follows:

Using the auxiliary expressions below,

$$T(y) = \frac{\alpha^2 + \beta^2}{Re} - \alpha\frac{3}{2}(1-y^2) \qquad (2\text{-}13)$$

$$I(y) = \frac{\alpha^2 + \beta^2}{Re} + T(y) \qquad (2\text{-}14)$$

$$J(y) = (\alpha^2 + \beta^2)T(y) - 3\alpha \qquad (2\text{-}15)$$

the fundamental vorticity equation becomes

$$[\frac{1}{Re} H^{iv} + I(y)H'' + J(y)H] - \gamma[H'' + (\alpha^2 + \beta^2)H] = 0. \qquad (2\text{-}16)$$

The associated vorticity equation is

$$[\frac{1}{Re} G'' + (T(y)-\gamma)G = \frac{\beta}{\alpha^2+\beta^2}[\frac{1}{Re} H''' + (T(y)-\gamma)H' - 3\alpha y H] . \qquad (2\text{-}17)$$

The number of independent parameters in equations 2-13 through 2-17 can be reduced to four by utilizing the

following functional transformations.

$$Re = \kappa^{-1} Re^{*} \tag{2-18}$$

$$\alpha^2 + \beta^2 = \alpha^{*2} \tag{2-19}$$

$$\alpha = \alpha^{*} \kappa e^{i\theta} \tag{2-20}$$

$$\gamma = \kappa \gamma^{*} \tag{2-21}$$

$$H(y) = \kappa^{-1} e^{-i\theta} H^{*}(y) \tag{2-22}$$

$$G(y) = \kappa^{-1} e^{-i\theta} \beta G^{*}(y) \tag{2-23}$$

Substitution of Eqs. 2-18 through 2-23 into the general solution, Eqs. 2-13 through 2-17 yields the three auxiliary functions

$$T^{*}(y) = \frac{\alpha^{*2}}{Re^{*}} - \alpha^{*} e^{i\theta} \, \frac{3}{2}(1-y^2) \tag{2-24}$$

$$I^{*}(y) = \frac{\alpha^{*2}}{Re^{*}} + T^{*}(y) \tag{2-25}$$

$$J^{*}(y) = \alpha^{*2} T^{*}(y) - 3\alpha^{*} e^{i\theta}. \tag{2-26}$$

The principal vorticity transport equation becomes

$$\left[\frac{1}{Re^*}H^{*iv}+I^*(y)H^{*''}+J^*(y)H^*\right] - \gamma^*\left[H^{*''}+\alpha^{*2}H^*\right] = 0 \qquad (2\text{-}27)$$

The associated vorticity transport equation becomes

$$\left[\frac{1}{Re^*}G^{*''}+(T^*(y)-\gamma)G^* = \frac{1}{\alpha^{*2}}\,\frac{1}{Re^*}H^{*'''}+(T^*(y)-\gamma^*)H^{*'}\right.$$

$$\left. -3\alpha^* e^{i\theta} yH^*\right] . \qquad (2\text{-}28$$

Equations 2-24 through 2-28 now involve only four parameters Re^*, α_R^*, α_I^* and θ. κ, the fifth parameter, has cancelled out. κ becomes part of the solution again during the reverse transformation of results from starred parameters to the original parameters.

III. STABILITY CRITERION

A. BACKGROUND

Studies of the stability of Poiseuille flow have used various criteria for determining the stability of the flow from the solutions obtained. The growth rate in time, γ_R, is usually used when there is no real-exponential spatial variation [Salven and Grosch, 1972]. When exponential growth in space has been included [Garg and Rouleau, 1971] the real part of the spatial wave number has been used to give the instability but this procedure, while seemingly plausible at first inspection, cannot be really justified with any rigor. The Lagrangian approach described below is believed to be a superior method for dealing with this case. In other cases, stability has arbitrarily been evaluated with respect to a frame of reference moving downstream at the phase velocity of the perturbation but again, this procedure has no strict rational justification, and especially so in connection with the fully three-dimensional perturbations considered in this thesis.

B. LAGRANGIAN REFERENCE

For perturbations that are both oscillatory and have exponential rates of growth or decay in the streamwise and transverse directions a Lagrangian frame of reference proves useful. The fluid particles have velocities varying

from 0 to 1.5 depending on their distance, y, from the walls. The velocity distribution is given by

$$U = \frac{3}{2}(1-y^2) \ . \tag{3-1}$$

Consider a coordinate system moving in the x direction with the mean velocity of a given fluid particle. Let y be the mean vertical coordinate of the moving particle. Then the velocity of the moving reference frame is the same as the velocity of the streamline along which the above particle moves and is given by Eq. 3-1. Let x', y, z, t be the coordinates and α, β, γ' the complex wave numbers with respect to the moxing axes. The form of the perturbation vector potential for a given eigenvalue obtained as a solution is

$$\overline{W} = \overline{j}G(y) + \overline{k}H(y) \exp(\alpha x + \beta z + \gamma t) \ . \tag{3-2}$$

The complex frequency γ' seen from this moving reference is different than from a fixed frame. To relate γ' to γ the perturbation vector potential is written in the moving frame and transformed into the form for the fixed frame.

$$\bar{W} = (\bar{j}G + \bar{k}H)\exp(\alpha x' + \beta z + \gamma' t)$$

$$= (\bar{j}G + \bar{k}H)\exp(\alpha(x-Ut) + \beta z + \gamma' t)$$

$$= (\bar{j}G + \bar{k}H)\exp(\alpha x + \beta z + (\gamma' - \alpha U)t)$$

$$= (\bar{j}G + \bar{k}H)\exp(\alpha x + \beta z + \gamma t) \tag{3-3}$$

Therefore

$$\gamma' - \alpha U = \gamma \tag{3-4}$$

Solving for γ' and splitting into real and imaginary parts yields

$$\gamma_R' = \gamma_R + \alpha_R U \tag{3-5}$$

$$\gamma_I' = \gamma_I + \alpha_I U \tag{3-6}$$

If γ_R' is positive, zero, or negative, the perturbation is said to be unstable, neutral, or stable, respectively, with respect to the moving reference frame. Thus, the value of γ_R' is taken to be a measure of the stability of each eigenvalue obtained.

C. TRANSFORMED STABILITY CRITERION

Consider now the transformation to the starred parameters. The condition of stability is determined by the value of γ_R' which Eq. 3-5 gives as

$$\gamma_R' = \gamma_R + \alpha_R U \ . \qquad (3\text{-}5)$$

Now

$$\gamma_R' = \kappa \gamma_R^{*'}$$

$$\gamma_R = \kappa \gamma_R^{*} \qquad (3\text{-}7)$$

$$\alpha_R = \kappa \alpha_R^{*}$$

Substitution of Eqs. 3-7 into Eq. 3-5 yields

$$\gamma_R^{*'} = \gamma_R^{*} + \alpha_R^{*} U \qquad (3\text{-}8)$$

α_R^{*} is defined by Eq. 2-10 as

$$\alpha_R^{*} = A^{*} \cos(\emptyset^{*} + \theta) \qquad (2\text{-}10)$$

Therefore

$$\gamma_R^{*'} = \gamma_R^{*} + A^{*} \cos(\emptyset^{*} + \theta) \qquad (3\text{-}9)$$

The three stability boundaries, incipient, critical, and fully developed are determined by the condition that exists when $\gamma_R^{*\prime} = 0$. Setting Eq. 3-9 equal to zero and solving for γ_R^* yields

$$\gamma_R^* = UA^* \cos(\phi^* + \theta) \ . \tag{3-10}$$

Now incipient instability corresponds to zero growth rate with respect to a coordinate system which moves with the flow velocity at the wall, which is zero.

$$\gamma_R^*)_I = 0 \tag{3-11}$$

Critical instability corresponds to zero growth rate with respect to a coordinate system which moves with the mean velocity of the flow, which is unity.

$$\gamma_R^*)_C = -A^* \cos(\phi^* + \theta) \tag{3-12}$$

Fully developed instability corresponds to zero growth rate with respect to a coordinate system which moves with the velocity of the flow on the centerline.

$$\gamma_R^*)_D = -\frac{3}{2} A^* \cos(\phi^* + \theta) \tag{3-13}$$

IV. RESULTS

A. TRANSFORMED PARAMETERS

Equations 2-24 through 2-28 were solved on the IBM-360 computer and the most unstable growth rate, γ_R^*, for a given $\emptyset^*$, θ, Re^*, A^* was obtained. The boundaries for incipient and critical instability were determined by the criteria explained in Section II. Fully developed instability did not occur for the case studies. Two values of $\emptyset^*$ were studied at various values of θ, Re^*, and A^*.

1. $\underline{\emptyset^* = 90°}$

Values of θ explored were 0, 1, 2, 3, 4, 5, and 6°. Because one-degree increments of θ yield graphical results that are extremely cluttered, this study will present the results only for θ = 0, 3, and 6°. Figure 4-1 shows the stability boundaries for $\emptyset^* = 90°$. The effect of changing θ, while holding $\emptyset^*$ constant, can be seen. An increase in θ causes a degrease in Re^* for both incipient and critical instability. Also, for a given θ, the boundary for critical instability occurs at a higher Re^* than does the boundary for incipient instability.

There is only one curve for $\theta = 0°$. In this case the criteria for incipient, critical, and fully developed instability turn out to be identical. Equation 3-10 shows

$$\gamma_R^* = -UA^*\cos(\emptyset^* + \theta) \qquad (3\text{-}10)$$

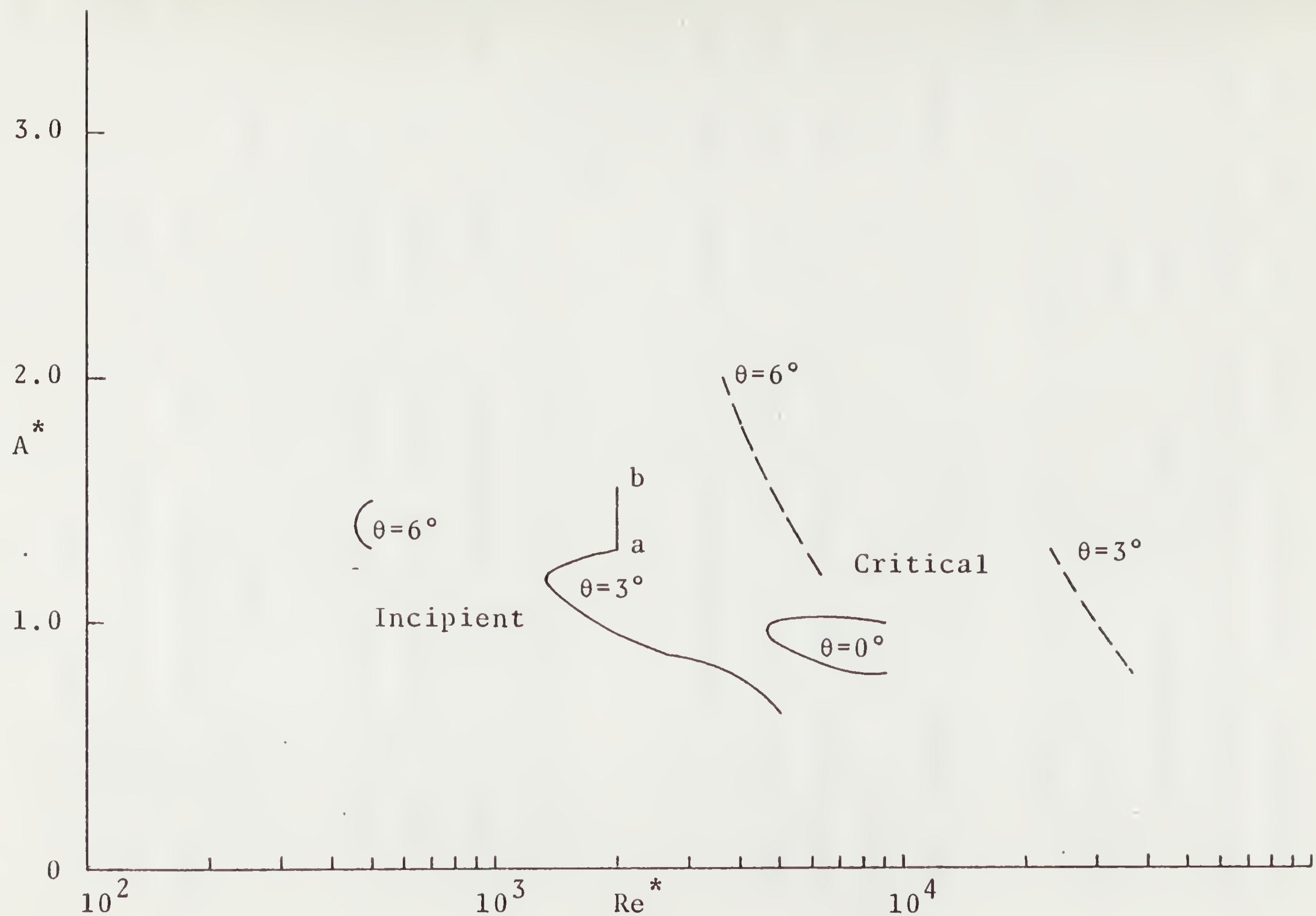

Figure 4.1. Transformed Stability Boundaries for ϕ^* = 90°

and cos(90° + 0°) = 0; thus the criterion for stability on the three boundaries is $\gamma_R^* = 0$.

Note that the boundary of incipient instability for $\theta = 3°$ shows an abrupt distoncinuity represented by segment ab. This discontinuity is similar to that obtained by Harrison. It is expected that extension of the incipient boundary for other values of θ would reveal the same characteristic.

Figures 4-2 and 4-3 are plots of the growth rate, γ_R^*, versus Re^* for $\theta = 3°$ and 6°, respectively. Both plots show the locus of points that represent the boundaries of incipient and critical instability. It can be seen that fully developed instability is not reached even at $Re^* = 100,000$.

2. <u>$\phi^* = 95°$</u>

Due to a lack of time only two values of θ were explored, $\theta = 0$ and 3°. A comparison of Fig. 4-4 with 4-1 shows that the stability contours follow much the same pattern for both cases. However, increasing ϕ^* to 95° causes a corresponding decrease in Re^*.

Figures 4-5 and 4-6 show the growth rate as a function of Re^* for $\theta = 0$ and 3°, respectively. The locus of points that represent the boundaries of incipient and critical instability can again be seen. Fully developed instability is not reached at $Re^* = 100,000$.

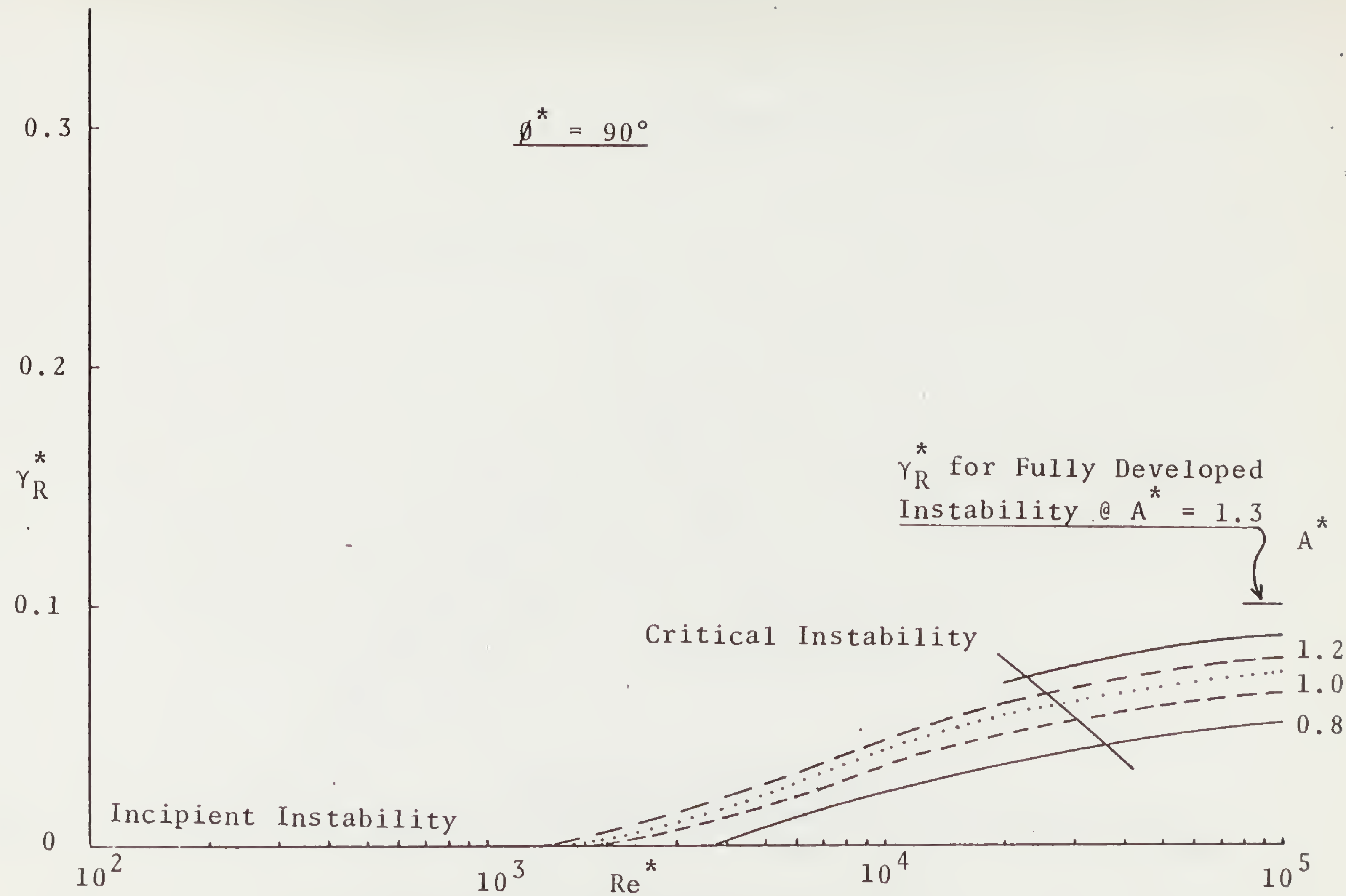

Figure 4-2. Growth Rate for $\theta = 3°$

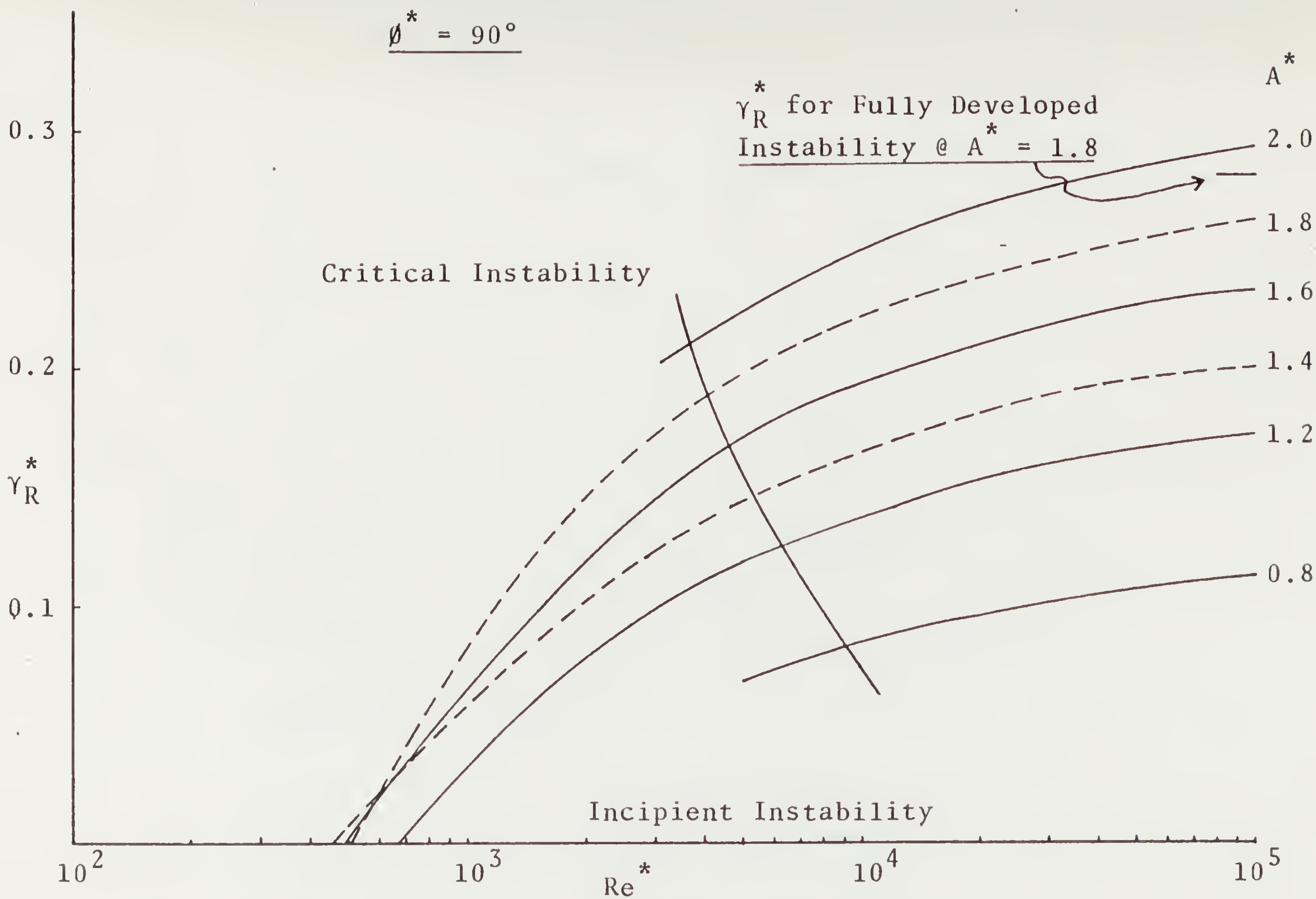

Figure 4-3. Growth Rate for $\theta = 6°$

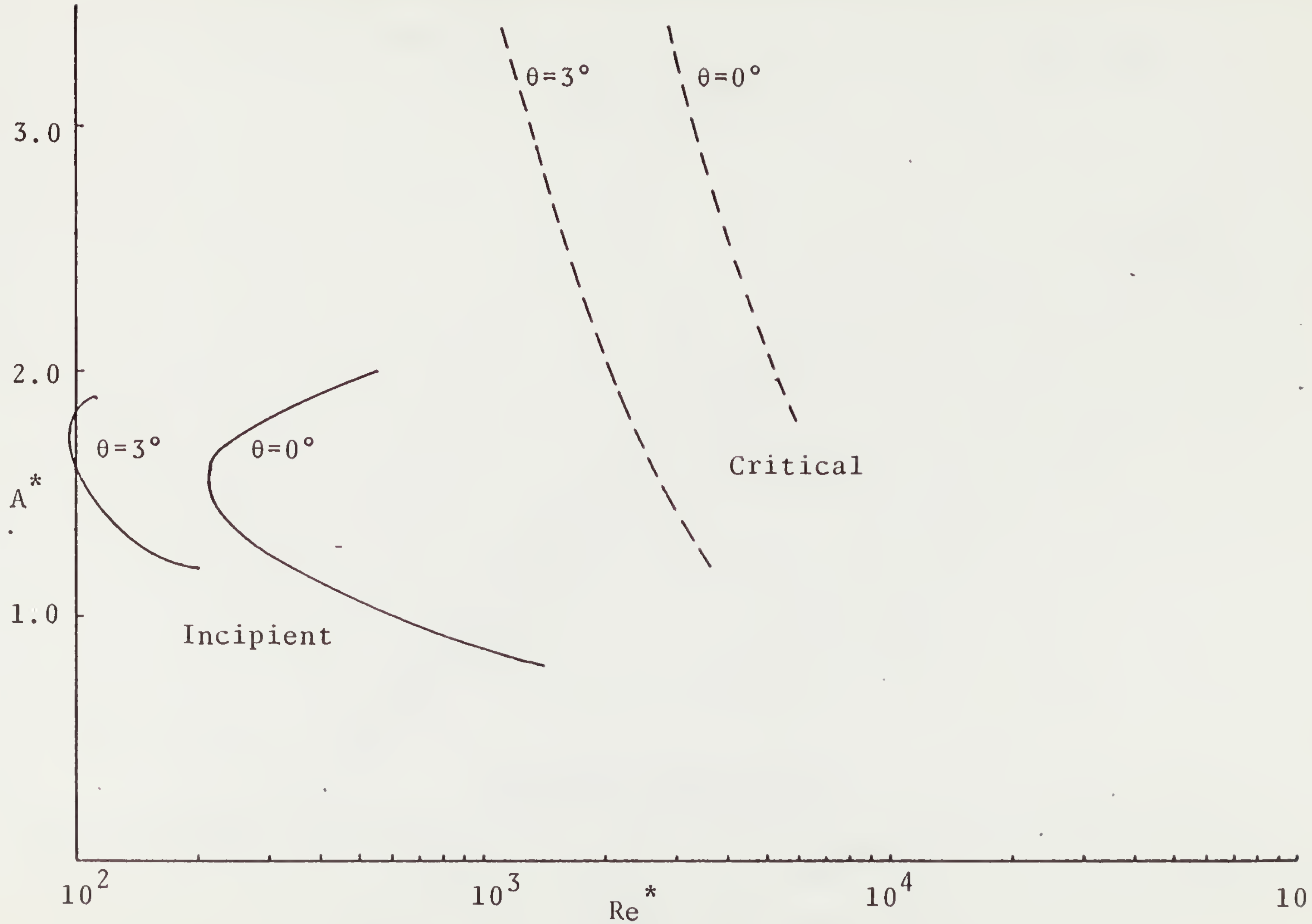

Figure 4-4. Transformed Stability Boundaries for $\emptyset^*$ = 95°

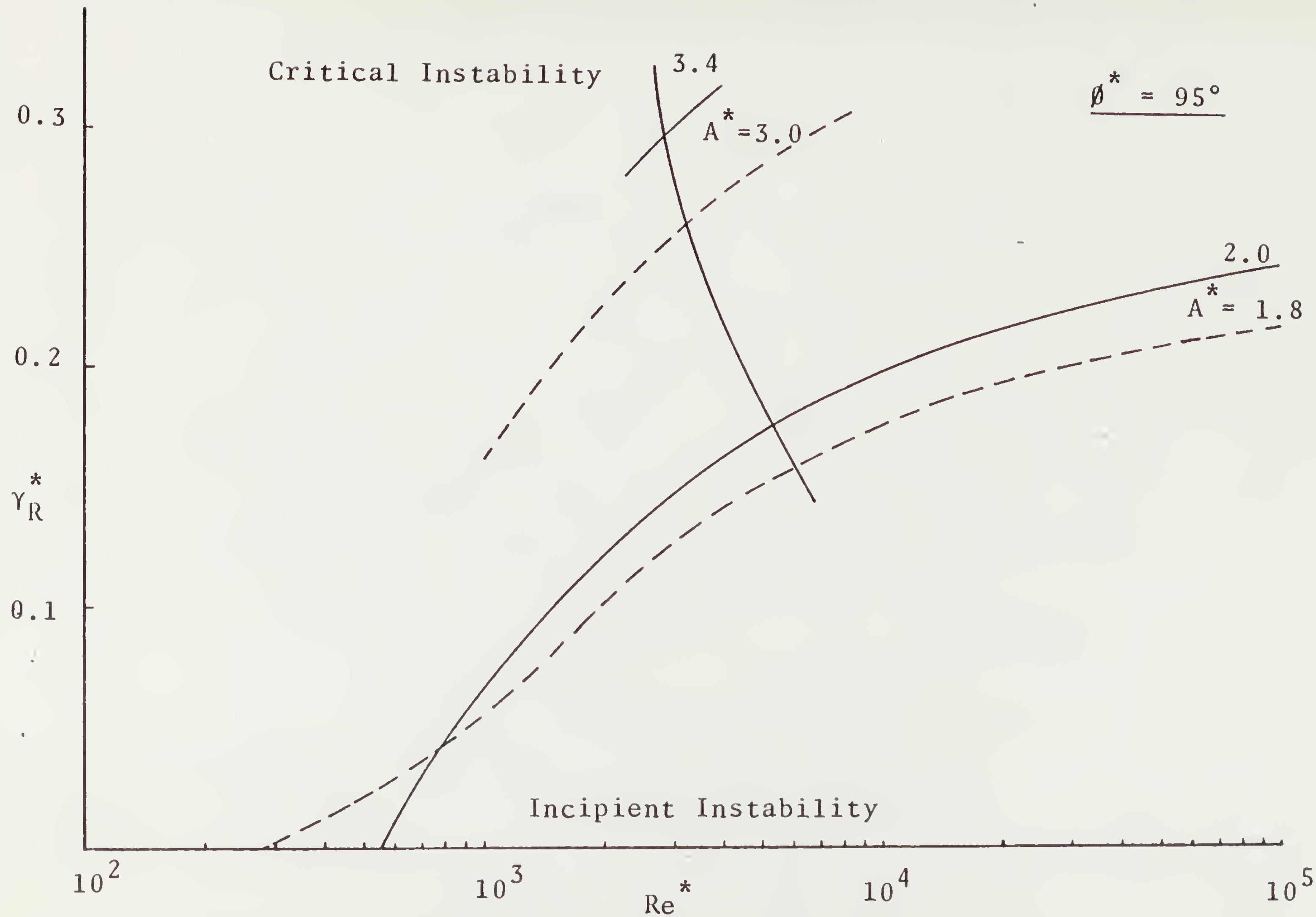

Figure 4-5. Growth Rate for $\theta = 0°$

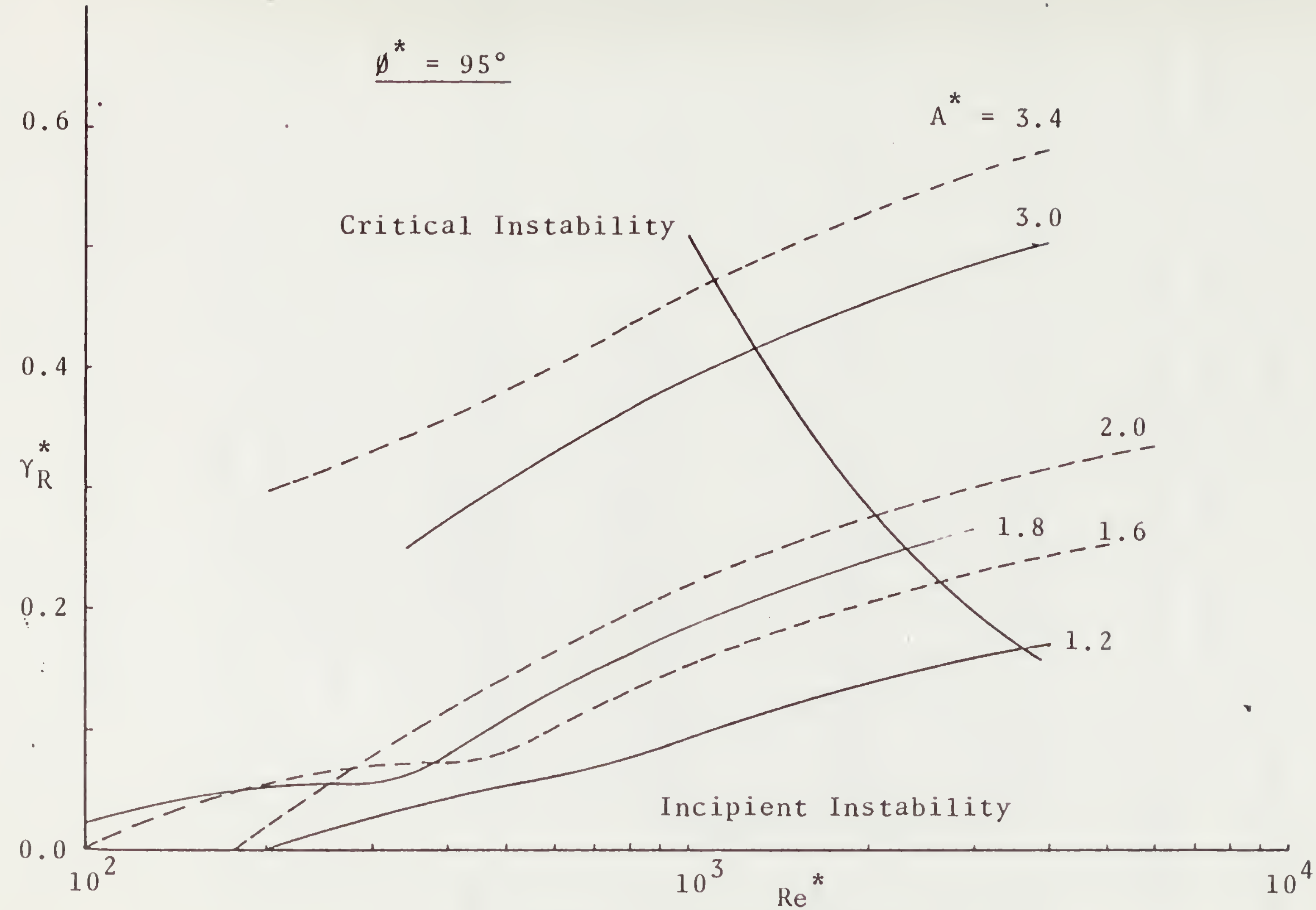

Figure 4-6. Growth Rate for θ = 3°

B. RESULTS TRANSFORMED TO UNSTARRED PARAMETERS

In order to present the results in a manner consistent with Ref. 1, it is necessary to transform the results from starred to unstarred parameters. This transformation puts the results in a more easily understandable form.

From Fig. 4-7 the following relations can be deduced.

$$\left(\frac{\lambda_R}{A^*}\right)^2 = 1/2(\kappa^2 + [(1-\kappa^2)^2 + 4\kappa^2\sin^2\theta]^{1/2} + \cos 2\phi^*) \tag{4-1}$$

$$\left(\frac{\lambda_I}{A^*}\right)^2 = 1/2(\kappa^2 + [(1-\kappa^2)^2 + 4\kappa^2\sin^2\theta]^{1/2} - \cos 2\phi^*) \tag{4-2}$$

$$\tan^2\Lambda_R = \frac{[(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}+\cos 2\phi^*-\kappa^2\cos 2(\phi^*+\theta)}{\kappa^2[1 + \cos 2\phi^*+\theta)]} \tag{4-3}$$

$$\tan^2\Lambda_I = \frac{[(1-\kappa^2)^2+4\kappa^2\sin^2\theta]^{1/2}-\cos 2\phi^*+\kappa^2\cos 2(\phi^*+\theta)}{\kappa^2[1 - \cos 2(\phi^*+\theta)]} \tag{4-4}$$

λ_R and Λ_R are the magnitude and direction, respectively, of the perturbation and growth vector. λ_I and Λ_I are the magnitude and direction, respectively, of the perturbation oscillation vector.

1. Perturbation Rate Vectors, Magnitude

Figure 4-8 shows G as a function of Re/Re^* with θ as an independent parameter; the curves are valid for all

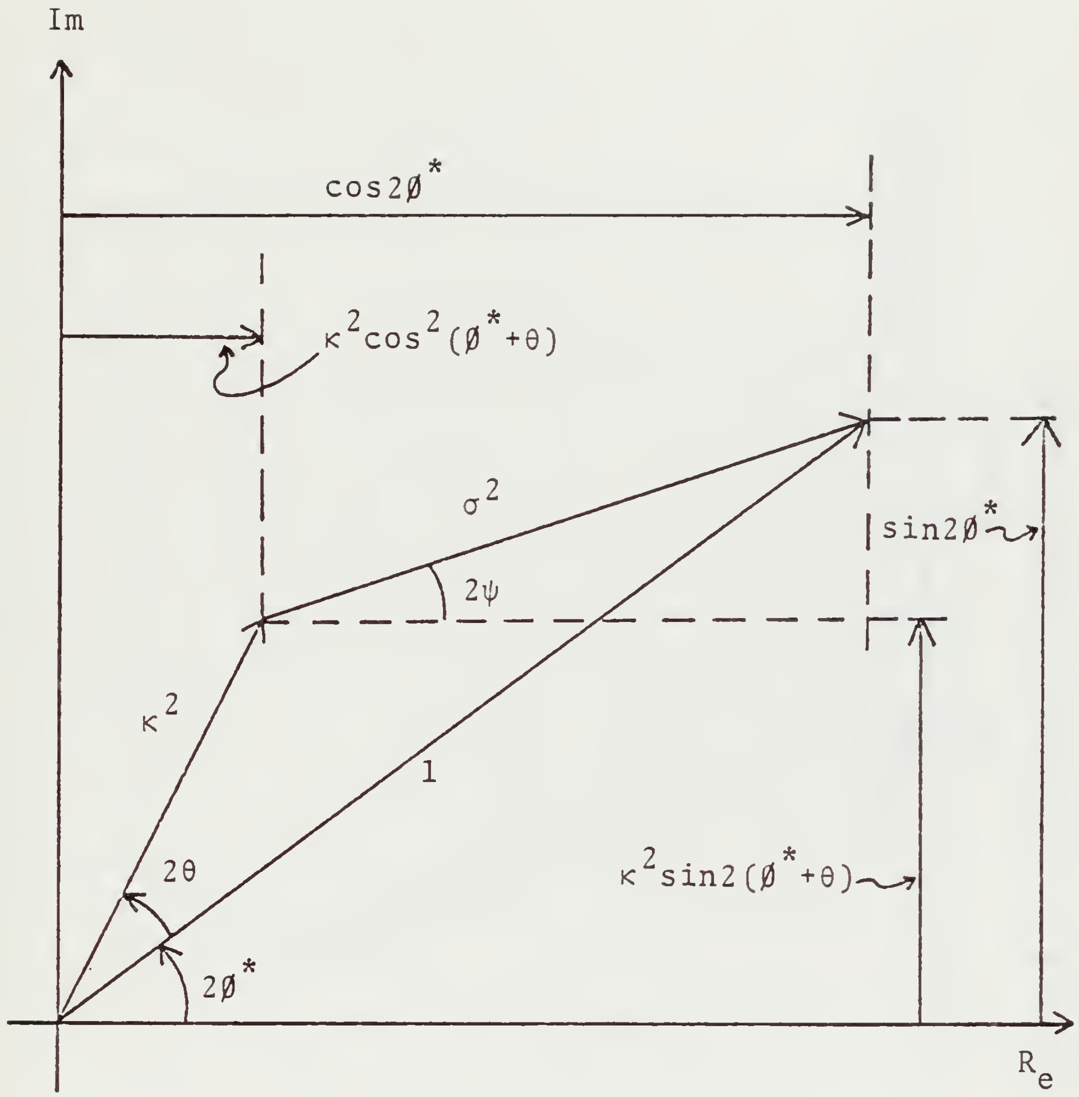

Figure 4.7. Vector Diagram for Parameter Transformation

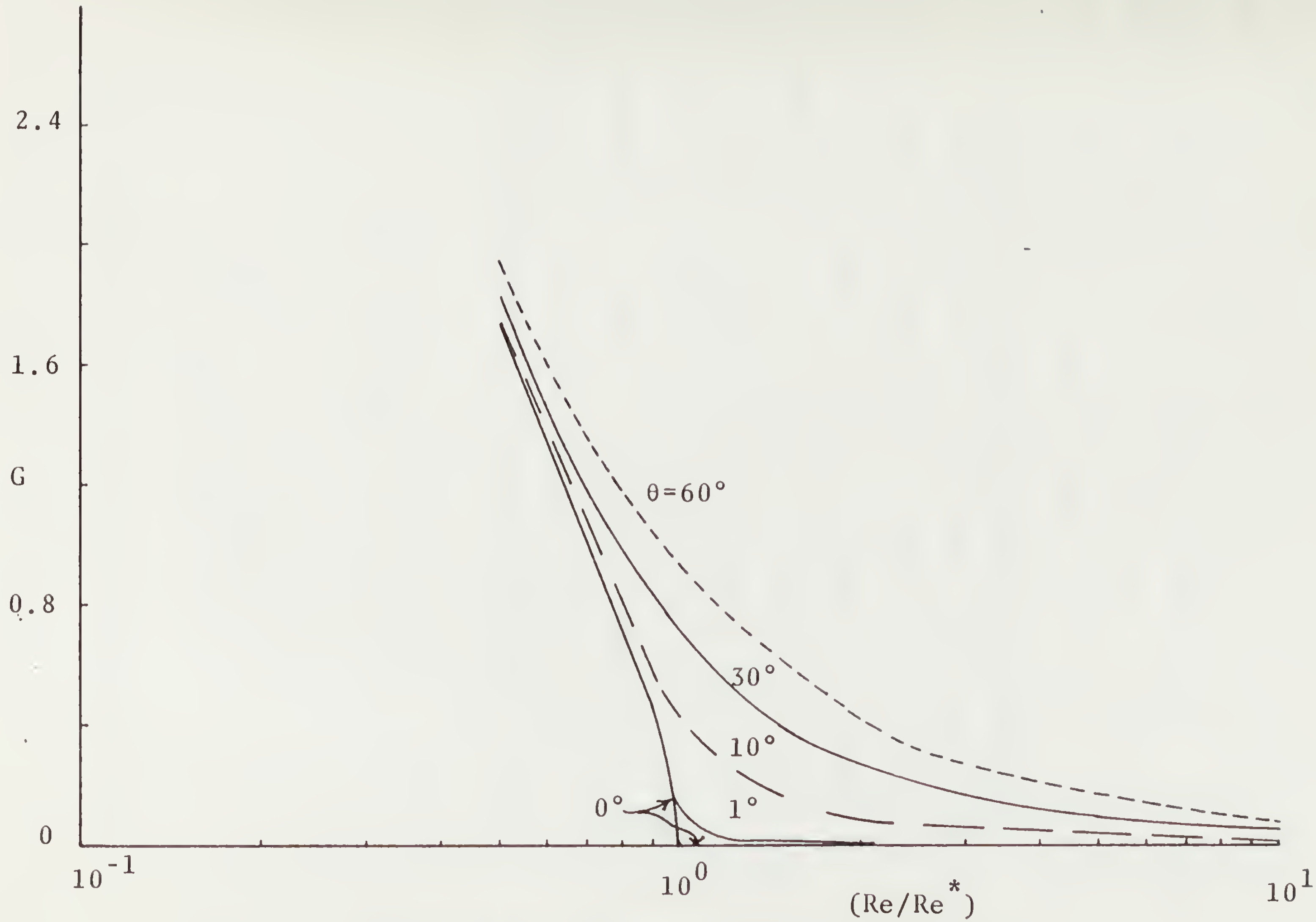

Figure 4-8. Parameter G versus Re/Re^*

values of $\emptyset^*$. For any fixed value of θ, G decreases as the Reynolds number ratio increases. In the case of $\theta = 0°$ G decreases with increasing Reynolds number ratio until $Re/Re^* = 1$. G then maintains a constant value of zero for further increases in the Reynolds number ratio. Although values of θ above 6° were not explored, values of θ up to 60° are shown here to demonstrate the trend as θ increases.

2. Perturbation Growth Rate Vectors, Direction

The quantities of $\tan^2\Lambda_R$ and $\tan^2\Lambda_I$ are fixed by Eqs. 4-2 and 4-3, respectively. However, if $\tan^2\Lambda_R$ be specified, this does not fix Λ_R uniquely as there are four angles, one in each quadrant, which have the specified value of $\tan^2\Lambda_R$. Similar considerations apply also to the other angle Λ_I. Hence, to determine Λ_R and Λ_I uniquely it is necessary to consult auxiliary relations which fix the quadrant in which these angles really fall.

The components of λ_R which fix angle Λ_R are

$$\alpha_R = \lambda_R \cos\Lambda_R = \kappa A^* \cos(\emptyset^* + \theta) \tag{4-5}$$

and

$$\beta_R = \lambda_R \sin\Lambda_R = \sigma A^* \cos\psi \ . \tag{4-6}$$

The components of λ_I which fix angle Λ_I are

$$\alpha_I = \lambda_I \cos\Lambda_I = \kappa A^* \sin(\emptyset^* + \theta) \tag{4-7}$$

and

$$\beta_I = \lambda_I \sin\Lambda_I = \sigma A^* \sin\psi \ . \qquad (4\text{-}8)$$

Moreover, in this study, the angle $(\emptyset^*+\theta)$ has been restricted to lie in the second quadrant so that

$$\alpha_R \leq 0 \qquad (4\text{-}9)$$

$$\alpha_I \geq 0 \qquad (4\text{-}10)$$

Recall that negative values of α_R were shown by Harrison to be destabilizing. That is why the present study is restricted to negative values of α_R.

In order to determine the algebraic signs of components β_R and β_I, it is necessary to bracket the range of the angle ψ. A study of Fig. 4-7 reveals that for positive values of θ, the following limits apply.

$$\lim_{\kappa \to 0} = \emptyset^* \qquad (4\text{-}11)$$

$$\lim_{\kappa \to 0} = (\emptyset^*+\theta) - \frac{\pi}{2} \qquad (4\text{-}12)$$

It is also evident that the x-y plane is a plane of symmetry and that therefore a reversal of the perturbation

characteristics with respect to the z axis is permissible and leaves the essential features of the solution otherwise unchanged. This amounts to saying that the angle ψ can always be changed by ±180°, with no significant effect upon the solution except for a reversal of the perturbations with respect to the plane of symmetry. For definiteness in this discussion, however, we limit the angle ψ as incicated by Eqs. 4-9 and 4-10. It is then evident that the above auxiliary relations, along with the basic relations of Eqs. 4-3 and 4-4 , now suffice to fix Λ_R and Λ_I uniquely for any assigned values of the parameters $\emptyset^*$, θ and κ.

Figure 4-9 shows Λ_R as a function of Re/Re^* for $\emptyset^* = 90°$ with θ as an independent parameter. This plot shows a constant value of $\Lambda_R = 90°$, for $\theta = 0°$ and $Re < Re^*$. When $Re > Re^*$, the vector magnitude, λ_R is zero. For all other values of θ the perturbation growth rate vector, $\overline{\lambda}_R$, rotates from near the transverse to near the upstream direction as Re/Re^* increases.

Figure 4-10 shows the rotation of the perturbation growth rate vector as Re/Re^* increases, for $\emptyset^* = 95°$. Note that when $\emptyset^* = 90°$, Λ_R varies between 90° and 180° whereas when $\emptyset^* = 90°$, Λ_R varies between 90° and 270°.

3. Perturbation Oscillation Rate Vectors, Direction

Figures 4-11 and 4-12 are similar plots showing the rotation of the oscillation rate vector with changing Re/Re^* for $\emptyset^* = 90°$ and 95°, respectively. Comparing

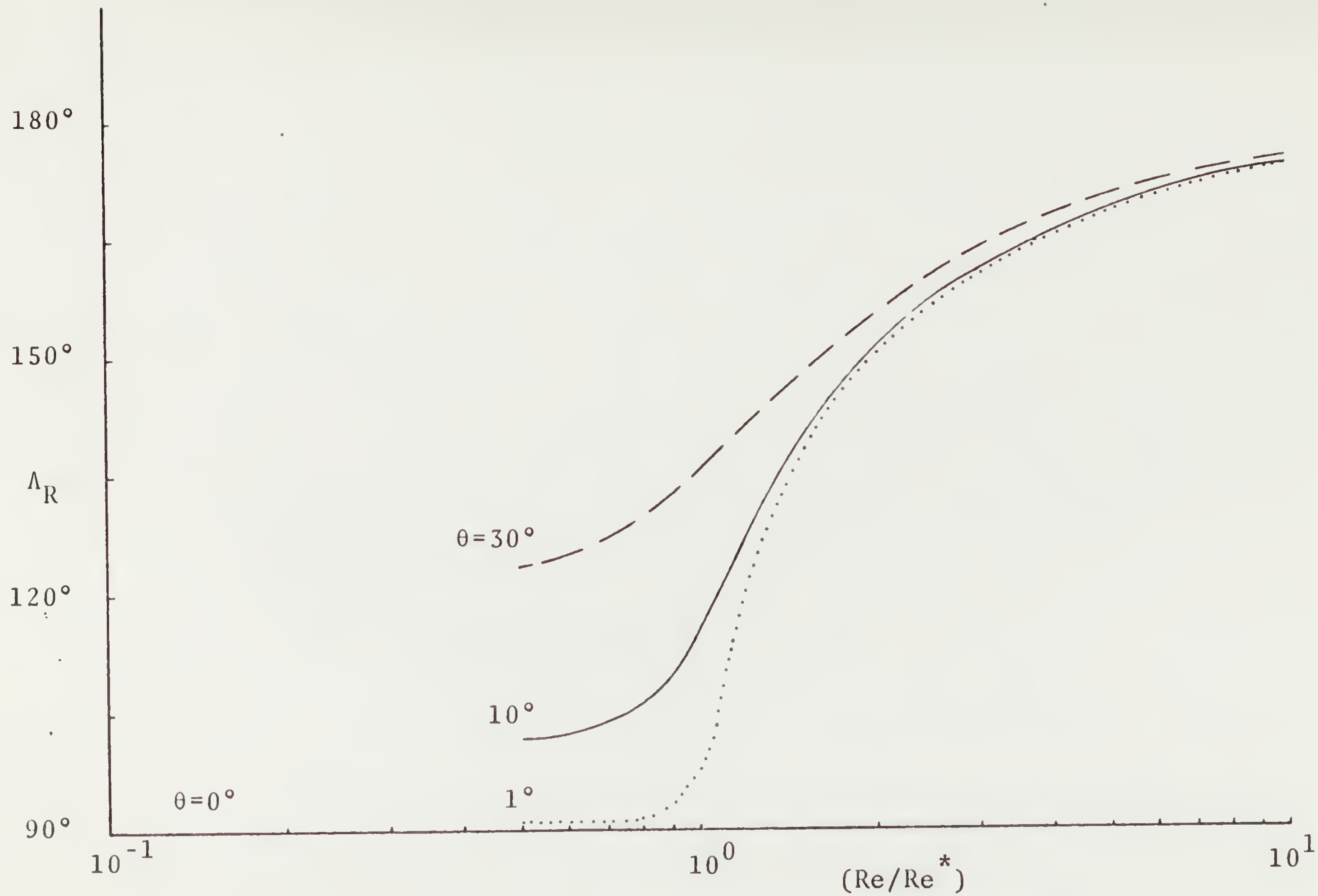

Figure 4-9. Direction of Growth Rate Vector for $\phi^* = 90°$

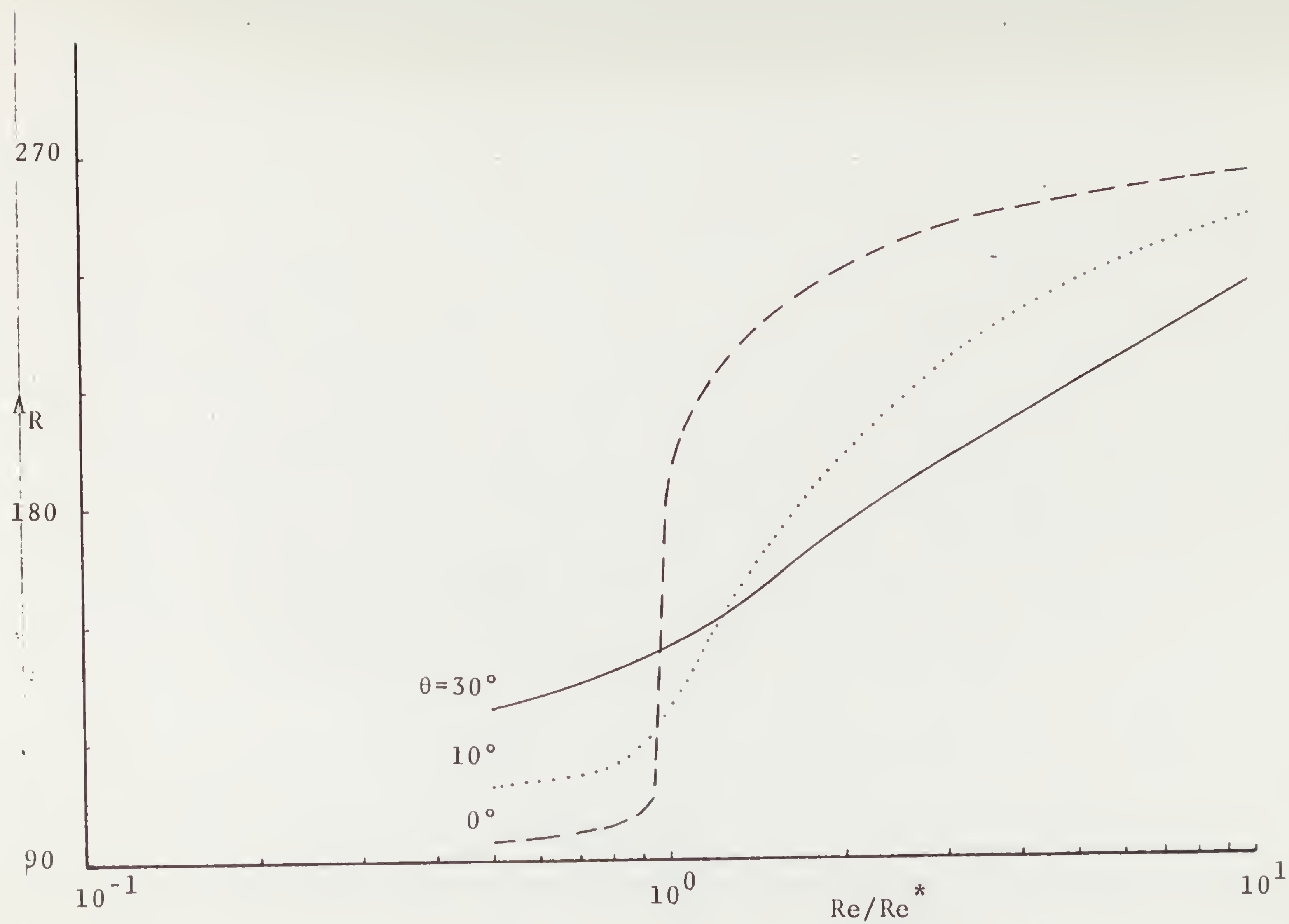

Figure 4-10. Direction of Growth Rate Vector for ϕ^* = 95°

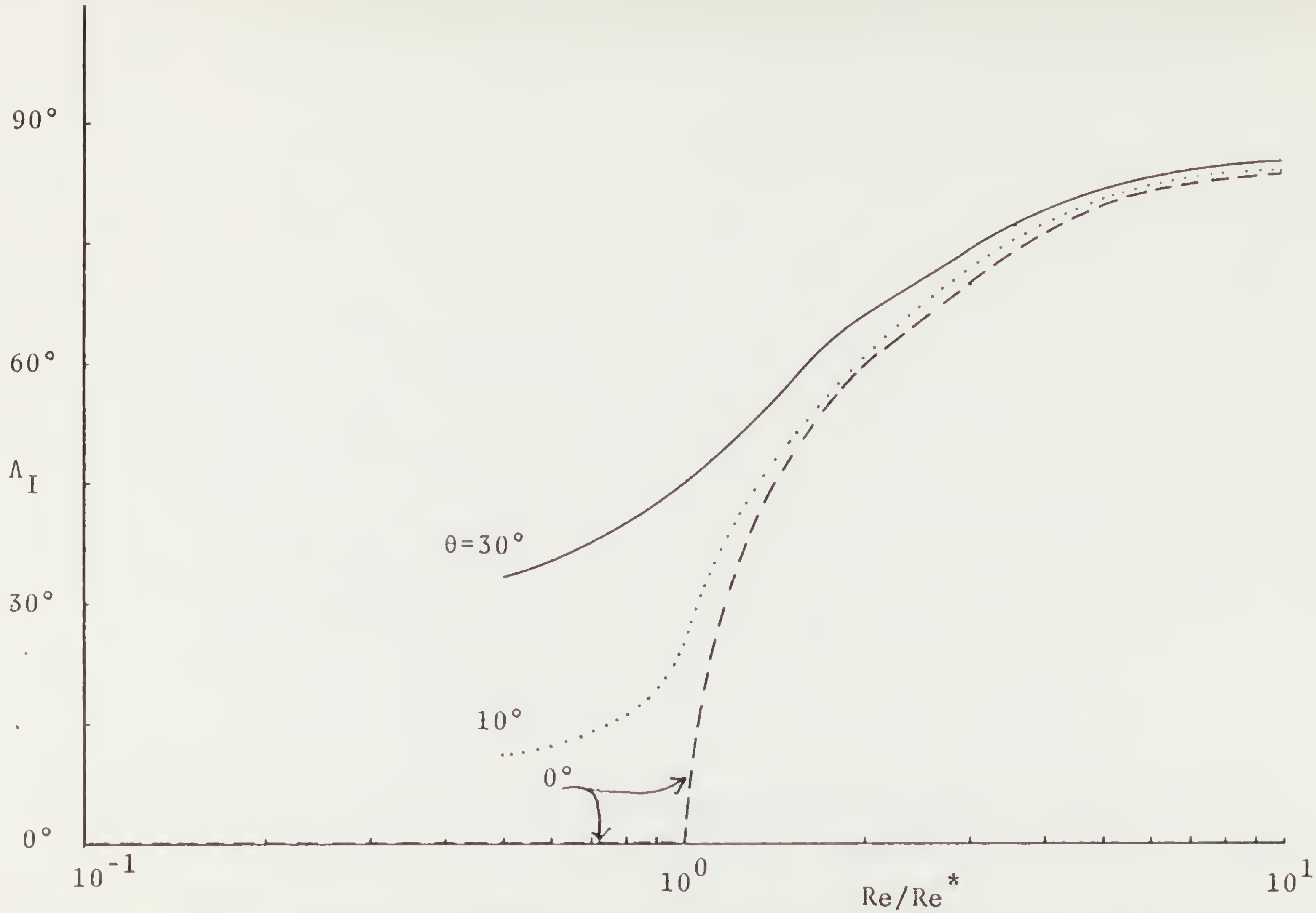

Figure 4-11. Direction of Oscillation Rate Vector for ϕ^* = 90°

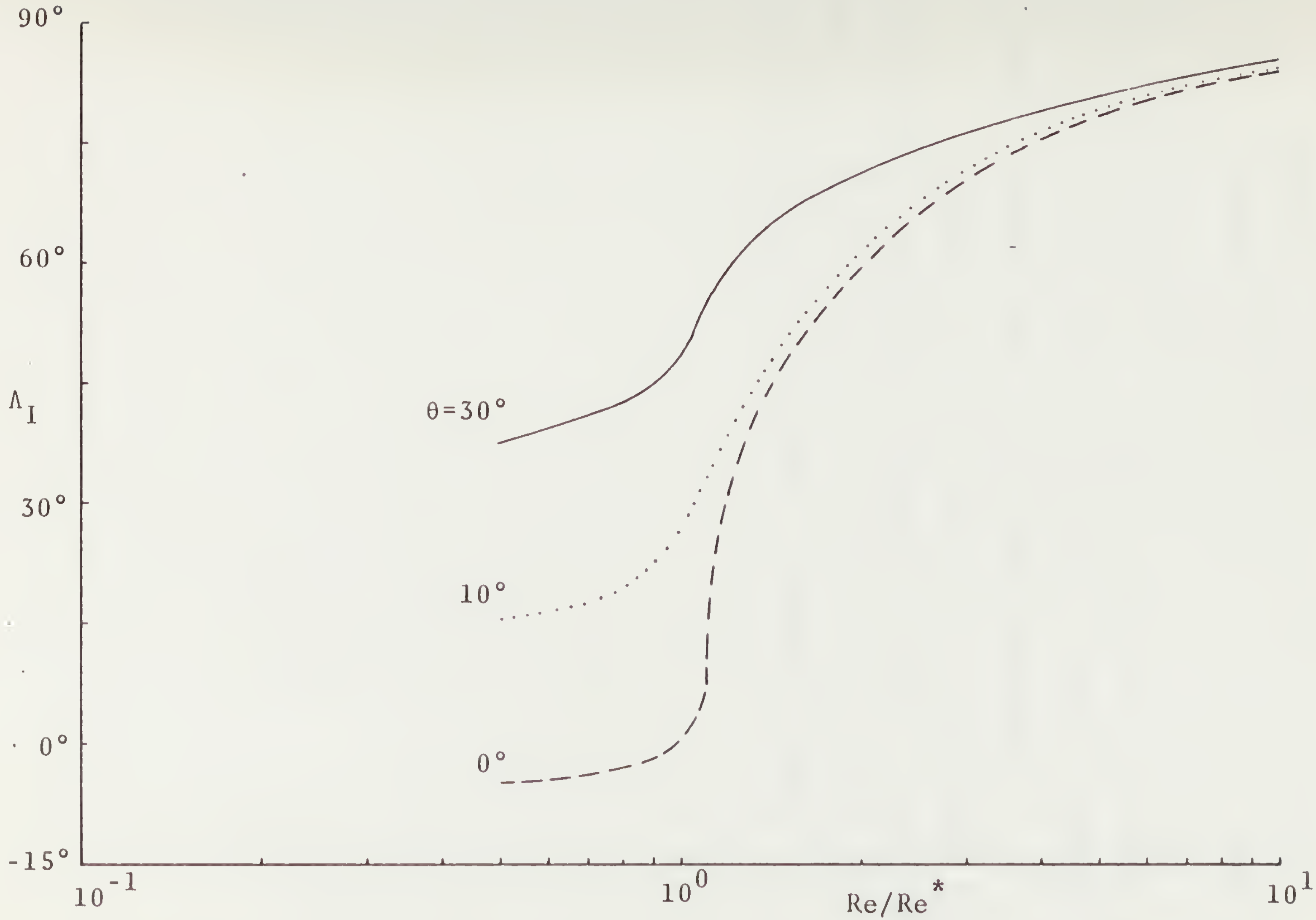

Figure 4-12. Direction of Oscillation Rate Vector for ϕ^* = 95°

Fig. 4-11 with 4-9, for $\emptyset^* = 90°$ and $\theta = 0°$ both Λ_R and Λ_I have constant values when $Re/Re^* <$ unity.

4. Stability Boundaries in Unstarred Parameters

Figures 4-13 and 4-14 show the stability boundaries for unstarred parameters. λ_R is 0.05 for both plots. A comparison with Figs. 4-1 and 4-4 shows that the character of the boundaries remains unchanged. However, Reynolds number is greater than starred Reynolds number.

Although it is not shown here, it was found that increasing λ_R causes the boundaries to move to the left and up. In other words, an increase in λ_R causes an increase in λ_I and a decrease in Reynolds number.

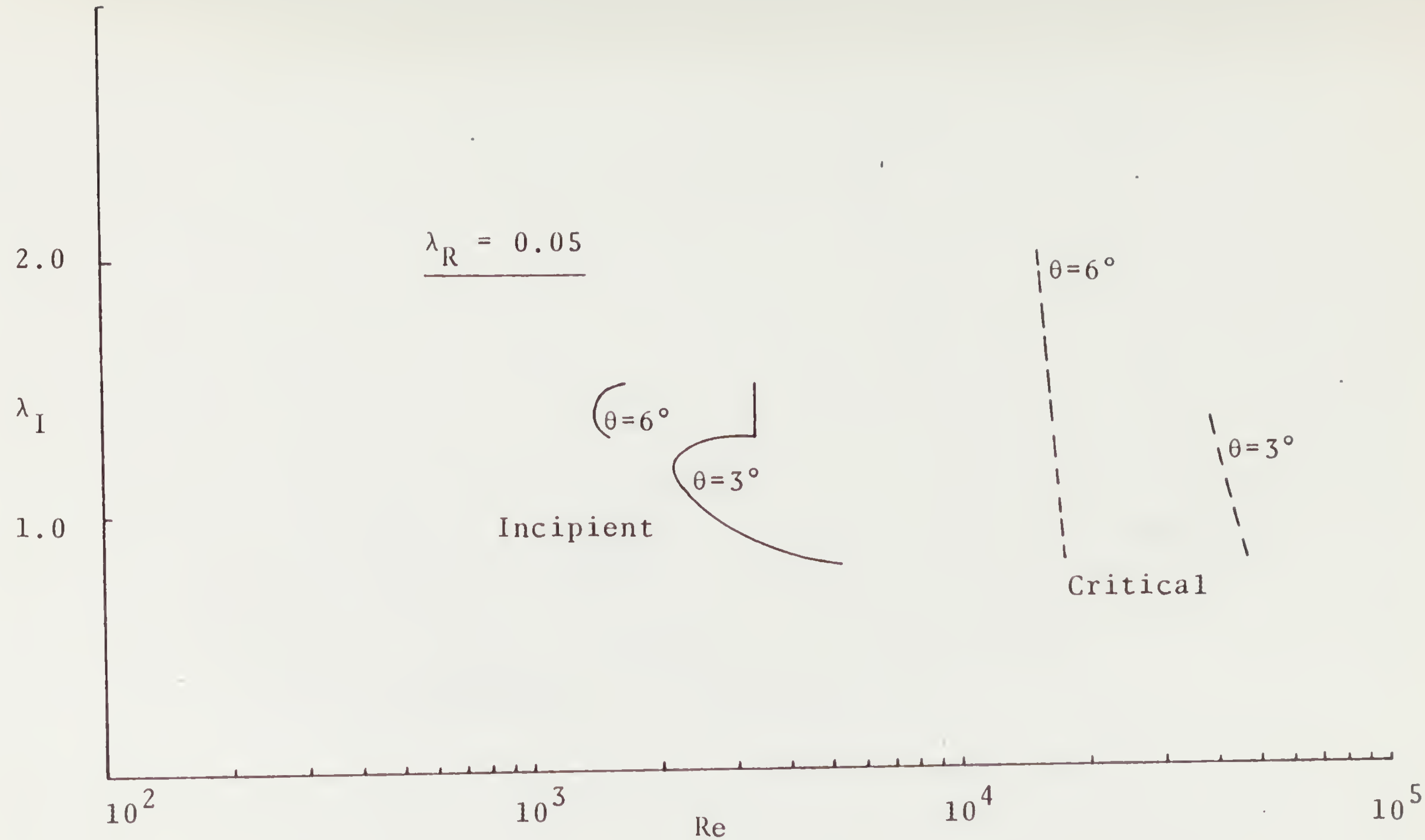

Figure 4-13. Stability Boundaries for ϕ^* = 90°

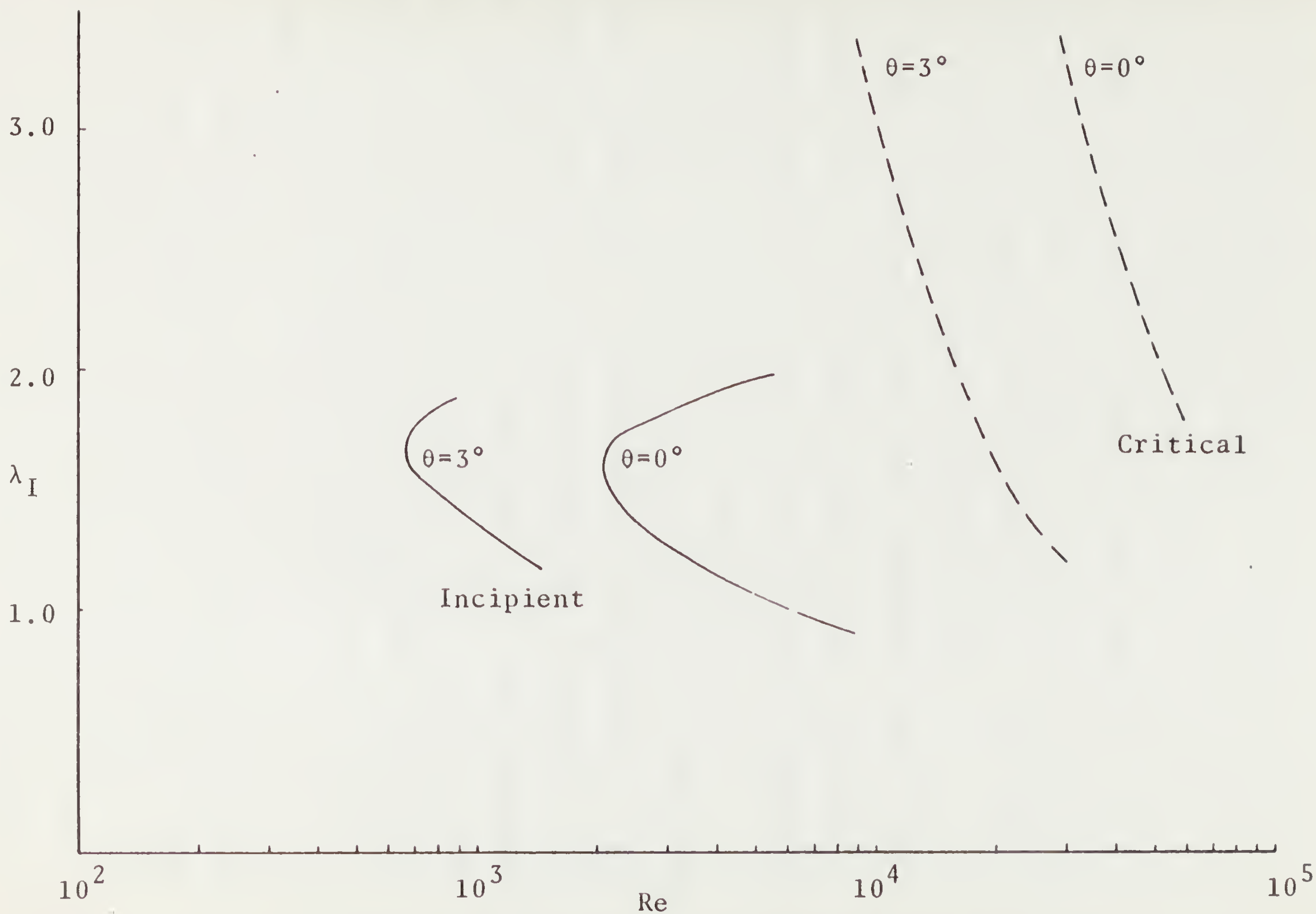

Figure 4-14. Stability Boundaries for $\phi^* = 95°$

V. CONCLUSIONS AND RECOMMENDATIONS

The research reported here is based largely on the theory developed by Harrison in Ref. 1. However, this theory is further extended in the present work by the introduction of a useful similarity transformation. The transformation reduces the number of independent parameters required in the computer solution from five to four and thereby significantly reduces computation time.

In Ref. 1, Harrison showed that negative values of α_R are destabilizing. The results presented here show that, for negative values of α_R, the critical Reynolds number for plane Poiseuille flow can be lowered indefinitely by proper selection of perturbation characteristics, provided that the corresponding increase in λ_R is acceptable. Thus the stability boundaries are not absolute in character but depend significantly on the "abruptness" of the perturbation in space as measured by parameter λ_R. The lowering of the critical Reynolds number in this way provides one possible explanation for the disagreement between earlier theory and experiment.

The stability of the flow along a particular streamline has been shown to depend on its velocity. Negative values of α_R yield the greater instabilities and streamlines with the lowest velocities, those nearest the walls, will be the most unstable. This seems to agree with experiment.

According to Schlichting, transition from laminar to turbulent flow is "characterized by an amplification of the initial disturbances and by the appearance of self-sustaining flashes which emanate from fluid layers near the wall along the tube."

Instability was found to be progressive in nature and two of the three defined stability boundaries were located, incipient and critical. Further research needs to be done for a wider range of parameters A^*, ϕ^*, and θ to find the combinations that correspond to minimum Reynolds numbers at the above stability levels.

APPENDIX A

USE OF THE COMPUTER PROGRAM

It was found extremely useful to precompile the program on a disk thereby avoiding the inconveniences of reading in the complete card deck for each run. An additional advantage is a reduction in turn-around time of considerable magnitude. The following will give procedures and hints that can be found in the W. R. Church Computer Center but require time-consuming search.

1. Pre-compiling Program

To compile the program, the following was read into the system

```
// Green Job Card,Time =(0,59)
// EXEC FORTCL
// FORT.SYSIN DD *
/*
          Program Card Deck goes here.  (no data)
//LINK.SYSLMOD DD DSNAME-S2593.LIB(POIS),DISP=(NEW,KEEP)
// UNIT=2321,VOLUME=SER=CEL006,LABEL=RETPD=220,
// SPACE=(CYL,(6,1,1),RLSE)
/*
```

2. Program Execution

Once the program is compiled, running the program consists of punching data cards in the namelist format and reading them in with the following deck of cards.

```
// Green Job Card,Time=(0,59)
// GO EXEC PGM=POIS,REGION=178K
//STEPLIB DD DSNAME=S2593.LIB,DISP=SHR,
// VOLUME=SER=CEL006,UNIT=2321
//FT06F001 DD sysout=A,DCB=(RECFM=FBA,LRECL=133,blksize=3325)
//FT05F001 DD *
&LIST N=30,REY=3000,TH=.052360,ASTAR=1.8,PHIS=95,&END
/*
```

Note: Column 1 is blank on the list card.

Three decks of these cards were used with each having a different job name, i.e., NEWBY 64A, NEWBY 64B, and NEWBY 64C. This proved useful as three jobs could be loaded at one time and one could keep track of what was already printed and what remained to be processed. It was also found to be useful to have three sets of job cards with each set having a different time. For Time=(0,59), 59 sec., one list card (data) was inserted. This was used for quick turn-around time and only a few points were being explored. For Time=(2,00), 2 minutes, three list cards could be read in and for Time=(4,00) six list cards could be used. Occasionally the four-minute time parameter would terminate execution after five list cards had been processed.

3. Program Alteration after Compilation

If changes were to be made in the program the file was scratched and a new file established with the changes

incorporated. To scratch the program on file the following deck was used.

```
// Green Job Card
// EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=2321,VOL=SER=CEL006,DISP=SHR
//SYSIN DD *
  SCRATCH VOL=2321=CEL006,DSNAME=S2593.LIB,PURGE
/*
```

Note: The scratch card begins in column 3.

In all three decks the name of the program (POIS) appears. The choice of a program name is an individual choice but once chosen it must appear the same in all card decks. The only other item that appears with uniqueness is the individual user number. In the context of this paper that number was 2593 and must agree with the user number on the job card.

There are two possible selections on input parameters for obtaining data. Both are used in this study. It is possible to select values of $\emptyset^*$, θ, Re^*, and vary A^* to find a point. This method was used first and worked well when obtaining a solution from the computer. The problem arises when interpreting and transforming the results. It proves useful to have values for fixed A^* and this method does not provide this easily.

An alternative technique is to select $\emptyset^*$, θ, A^*, and then vary Re^*. To construct Figures 3-1 through 3-6 the fixed A^* technique provides data in an easily usable form.

APPENDIX B

CHANGES TO COMPUTER PROGRAM IN REFERENCE 1

The following changes were made to the computer program in Ref. 1 to convert to starred parameters.

1. <u>Program #1</u>

Statement number 0002 (COMPLEX *16 A,B) was deleted and two type declaration statements (REAL*8 TH) and (COMPLEX*16 A) were inserted. The namelist statement, number 0008, was revised to read: NAMELIST / LIST /N,REY, TH,ASTAR,PHIS,VEL. Statement number 0018, B = DCMPLX(BR,BI) was deleted and the following statements added: PHI = PHIS/57.2958, AR = ASTAR * (DCOS(PHI)), AI = ASTAR *(DSIN (PHI)), THD = (TH*180.0)/3.141592654. Other changes to program #1 were those required to write out the revised inputs.

2. <u>Subroutine DEIGEO</u>

One small change was made to this subroutine due to the fact that values required by external functions CHM1E1 and CHM2E1 were passed by DEIGEO. Statement 0010 of DEIGEO, B = BETA, was deleted and TH = THETA was inserted. The common statement and type declaration statements were revised to incorporate the change to starred parameters.

3. <u>Functions CHM1E1 and CHM2E1</u>

External functions CHM1E1 and CHM2E1 required extensive modification as follows:

The type declaration statement (REAL*8 TH,DUR) was added.

CH4M1(Y) = A/REY was changed to CH4M1(Y) = A*EI/REY.

CH2M1(y) was changed to equal

```
-1.5DO*A**2*EI2*(1DO-y**2)+2DO*AEI*(A**2)/REY .
```

CHOM1(Y) was changed to equal

```
-AEI*((A**2)*(1.5DO*AEI*(1DO-Y**2)-(A**2)/REY)
+3DO*AEI)
```

CH2M2(Y) = A changed to CH2M2(Y) = AEI.

The following statements were added after CH2M2(Y) and ENTRY CHM2E1(k,Y):

```
DUR = 0.0
DU = DCMPLX(DUR,TH)
EI = CDEXP(DU)
AEI = A*EI
E12 = CDEXP(2*DU).
```

```
C.............................................................
C  PROGRAM #1
C
C                     PROGRAM TO PRINT EIGENVALUES
C                  FOR THE 3-D POISEULLE FLOW PROBLEM
C
C
C.............................................................
C
C      THIS PROGRAM SOLVES THE LINEARIZED NAVIER STOKES
C      EQUATION FOR POISEULLE FLOW.  THE EIGENVALUES
C      RESULTING FROM THE FINITE DIFFERENCE APPROXIMATION
C      ARE PRINTED.
C
C      INPUT
C
C          THE FOLLOWING MAY BE INPUT TO THE PROGRAM AS DATA
C          USING NAMELIST, 'LIST'.  NOTE, THE DEFAULT VALUES
C          ARE ONLY SET INITIALLY AND VALUES SET BY THE
C          USER WILL NOT BE CHANGED BETWEEN RUNS
C
C          N - HALF OF THE NUMBER OF FINITE DIFFERENCE GRID
C          POINTS ACROSS THE CHANNEL NOT INCLUDING THE END
C          POINTS.  N MUST BE .LE. MDIM, WHICH IS THE
C          DIMENSION OF THE MATRICES IN THIS PROGRAM.
C          DEFAULTED TO THE VALUE OF NDIM, THAT IS THE
C          DIMENSION OF THE MATRICES.  SEE PROGRAM BELOW FOR
C          THE DEFAULT VALUE.
C
C          REY - THE * REYNOLDS NUMBER (REAL*8) DEFAULT
C          VALUE = 6000.0
C
C          AR,AI - THE REAL AND IMAGINARY PARTS OF
C          THE STARRED WAVE NUMBERS  (REAL*8)
C          DEFAULTED TO 0.0 AND 1.0 RESPECTIVELY
C
C          VEL - THE VELOCITY OF THE MOVING COORDINATE
C          REFERENCE SYSTEM FOR WHICH THE STABILITY IS
C          DETERMINED.  (REAL*8)  DEFAULTED TO 0.0
C
C      OUTPUT
C
C          THE OUTPUT OF THIS PROGRAM IS A TABULATION OF THE
C          EIGENVALUES.  TWO LISTS ARE PRINTED, ONE FOR THE
C          EIGENVALUES CORRESPONDING TO EVEN EIGENFUNCTIONS
C          AND ONE FOR THOSE CORRESPONDING TO ODD EIGEN-
C          FUNCTIONS.  THE STABILITY OF EACH EIGENVALUE IS
C          PRINTED AND THE LEAST STABLE EIGENVALUE IS MARKED
C          WITH ASTERISKS.  A PLOT OF THE EIGENVALUES IS ALSO
C          PRINTED.
C
C      SUBROUTINES
C
C          THIS PROGRAM CALLS THE SUBROUTINE 'DEIGEO' TO
C          SOLVE FOR THE EIGENVALUES.  SUBROUTINE 'PLCTP'
C          IS USED TO PLOT THE EIGENVALUES ON THE PRINTER.
C
C
C.............................................................
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 TH
      COMPLEX*16 A
      REAL*4 GR4(60),GI4(60)
      REAL*8 GRE(60),GIE(60),GRO(60),GIO(60)
      COMPLEX*16 XMAT(60,30,3)
         COMPLEX*16 YMAT(60,30),WVEC(60),BMAT(5,60)
         EQUIVALENCE(YMAT(1,1),XMAT(1,1,3)),
     *               (BMAT(1,1),XMAT(1,1,3)),
     *               (WVEC(1),XMAT(1,6,3))
      NAMELIST / LIST / N,REY,TH,ASTAR,PHIS,VEL
C
```

```
C INITIALIZE VARIABLES  (SET DEFAULT VALUES)
C
      MDIM = 60
      N = 60
      REY = 6000D0
      TH = 0D0
      VEL = 0D0
C
C READ NAMELIST AND SET ALPHA AND ASTAR
C
    1 READ(5,LIST,END=100)
      PHI = PHIS/57.2958
      AR = ASTAR *(DCOS(PHI))
      AI = ASTAR * (DSIN(PHI))
      A = DCMPLX(AR,AI)
      THD = (TH*180.0)/3.141592654
C
C PRINT INPUT VALUES AS PAGE HEADING FOR EIGENVALUE LIST
C
      WRITE(6,9004)
 9004 FORMAT('1')
      WRITE(6,250) PHIS
  250 FORMAT('0',' PHI STAR =',2X,F10.7)
      WRITE(6,9005) N,REY,A,THD,VEL
 9005 FORMAT(' N =',I4,/,' REY =',F10.2,8X,'ALPHA =',
     *    2F12.7,8X,'THETA =',F12.7,/,'VEL =',F7.2)
      WRITE(6,9055)ASTAR
 9055 FORMAT('0','A-STAR =',F12.7)
C
C CALL SUBROUTINE TO SOLVE FOR EIGENVALUES.
C
      CALL DEIGEO(A,TH,REY,N,MDIM,GRE,GIE,GRO,GIO,XMAT,YMAT,
     *    BMAT,WVEC)
C
C DETERMINE WHICH EIGENVALUE IS THE LEAST STABLE.
C
      TEMP = -1D10
      MARK = 1
C
      DO 20 I=1,N
      IF(GRO(I)+AR*VEL.LT.TEMP) GO TO 20
      TEMP = GRO(I)+AR*VEL
      ITEMP = I
   20 CONTINUE
C
      DO 40 I=1,N
      IF(GRE(I)+AR*VEL.LT.TEMP) GO TO 40
      TEMP = GRE(I)+AR*VEL
      ITEMP = I
      MARK = 2
   40 CONTINUE
C
C LIST EIGENVALUES FOR ODD EIGENFUNCTIONS
C
      WRITE(6,9003)
 9003 FORMAT(///,6X,'GAMMA REAL',5X,'GAMMA IMAG',12X,'STAB')
      WRITE(6,9006)
 9006 FORMAT('0EIGENVALUES FOR ODD EIGENVECTORS',/)
C
      DO 50 I=1,N
      TEMP = GRO(I)+AR*VEL
      WRITE(6,9000) GRO(I),GIO(I),TEMP
      IF(I.EQ.ITEMP.AND.MARK.EQ.1) WRITE(6,9001)
   50 CONTINUE
 9000 FORMAT('0',1P2D15.4,1PD20.4)
 9001 FORMAT('+',52X,'***')
C
C LIST EIGENVALUES FOR EVEN EIGENVECTORS.
C
      WRITE(6,9007)
 9007 FORMAT('0EIGENVALUES FOR EVEN EIGENVECTORS',/)
      DO 55 I=1,N
```

```
      TEMP = GRE(I)+AR*VEL
      WRITE(6,9000) GRE(I),GIE(I),TEMP
      IF(I.EQ.ITEMP.AND.MARK.EQ.2) WRITE(6,9001)
   55 CONTINUE
C
C PUT EIGENVALUES INTO SINGLE PRECISION VECTORS TO PASS TO
C SUBROUTINE TO DO PLOTTING FOR ODD FUNCTIONS.
C
      DO 60 I=1,N
      GR4(I) = SNGL(GRO(I))
   60 GI4(I) = SNGL(GIO(I))
      WRITE(6,9004)
      CALL PLOTP(GR4,GI4,N,0)
      WRITE(6,9005) N,REY,A,THD,VEL
      WRITE(6,9055) ASTAR
C
C SIMILARLY PLOT EIGENVALUES FOR EVEN EIGENFUNCTIONS.
C
      DO 65 I=1,N
      GR4(I) = SNGL(GRE(I))
   65 GI4(I) = SNGL(GIE(I))
      WRITE(6,9004)
      CALL PLOTP(GR4,GI4,N,0)
      WRITE(6,9005) N,REY,A,THD,VEL
      WRITE(6,9055) ASTAR
C
      GO TO 1
  100 WRITE(6,9004)
      STOP
      END
```

```
C.....................SUBROUTINE DEIGEO.........................
C
C
C     PURPOSE
C
C        DEIGEO SOLVES THE LINEARIZED NAVIER-STOKES EQUATION
C        FOR POISEULLE FLOW.  THE INPUTS TO DEIGEO ARE THE
C        STARRED WAVE NO., ALPHA, THETA *, AND STARRED
C        REYNOLDS NUMBER.
C        DEIGEO OUTPUTS THE EIGENVALUES FOR GAMMA.
C
C     USAGE
C
C        CALL DEIGEO(ALPHA,THETA,REYNO,N,MDIM,WREVEN,WIEVEN,
C          WRODD,WIODD,CDM,DM,BM,WV)
C
C     DESCRIPTION OF PARAMETERS
C
C        THE FOLLOWING MUST BE SET BY THE CALLING PROGRAM...
C          ALPHA,THETA,REYNO,N,MDIM
C
C        ALPHA - THE PERTURBATION WAVE NUMBER IN THE FLOW
C          DIRECTION (X).  (COMPLEX*16)
C
C        REYNO - THE REYNOLDS NUMBER (REAL*8)
C
C        N - THE SIZE OF THE MATRICES WHICH IS EQUAL TO
C          (ND-1)/2 WHERE ND IS THE NUMBER OF DIVISIONS
C          ACROSS THE CHANNEL.  (NOTE... DEIGEO SOLVES THE
C          PROBLEM ACROSS THE HALF CHANNEL TWICE - ONCE FOR
C          THE EIGENVALUES CORRESPONDING TO THE EVEN
C          EIGENFUNCTIONS AND ONCE FOR THOSE CORRESPONDING
C          TO THE ODD EIGENFUNCTIONS.
C
C        MDIM - THE COLUMN DIMENSION OF THE MATRICES WHICH
C          DEIGEO USES.  MDIM MUST BE .GE. N
C
C        THE FOLLOWING ARE OUTPUT BY DEIGEO
C          WREVEN,WIEVEN,WRODD,WIODD
C
C        WREVEN,WIEVEN - THE REAL AND IMAGINARY PARTS OF THE
C          EIGENVALUES CORRESPONDING TO THE EVEN
C          EIGENFUNCTIONS.  DIMENSIONED TO AT LEAST N.
C          (REAL*8)
C
C        WRODD,WIODD - THE REAL AND IMAGINARY PARTS OF THE
C          EIGENVALUES CORRESPONDING TO THE ODD
C          EIGENFUNCTIONS.  DIMENSIONED TO AT LEAST N.
C          (REAL*8)
C
C        THE FOLLOWING MATRICES MUST BE INPUT TO DEIGEO AS
C          WORKSPACE.
C
C          CDM(MDIM,MDIM)  (COMPLEX*16)
C          DM(MDIM,MDIM)  (REAL*8)
C          BM(5,MDIM)  (COMPLEX*16)
C          WV(MDIM)  (COMPLEX*16)
C
C     NOTES...
C
C        THE MATRICES CAN BE OVERLAPPED TO CONSERVE SPACE,
C        FOR EXAMPLE, FOR N = 60...
C
C              COMPLEX*16 CDM(60,30,3)
C              COMPLEX*16 DM(60,30),WV(60),BM(5,60)
C              EQUIVALENCE (DM(1,1),CDM(1,1,3)),
C                          (BM(1,1),CDM(1,1,3)),
C                          WVEC(1),CDM(1,6,3))
C
C        NOTE THAT IT IS ONLY THE ACTUAL SIZE OF THESE
C        WORKSPACES THAT IS IMPORTANT, NOT THEIR TYPE.
C
```

```
C      OTHER ROUTINES NEEDED
C
C         THE FOLLOWING ARE CALLED BY DEIGEO
C
C            CHM1E1,CHM2E2,MSET,CDMTIN,BMSET,MULDBM,DSPLIT,
C            EHESSC,ELRH1C
C
C.............................................................
C
      SUBROUTINE DEIGEO(ALPHA,THETA,REYNO,N,MDIM,
     *     WREVEN,WIEVEN,WRODD,WIODD,CDM,DM,BM,WV)
      IMPLICIT COMPLEX*16(A-H,O-Z)
      DIMENSION IVEC(100)
      REAL*8 WREVEN(1),WIEVEN(1),WRODD(1),WIODD(1)
      REAL*8 CDM(1),DM(1),BM(1),WV(1)
      REAL*8 REY,DELY,REYNO
      REAL*8 TH,THETA
      COMMON / COEFNT / A,TH,G,REY,DELY
      EXTERNAL CHM1E1,CHM2E1
C
C THIS SUBROUTINE SOLVES THE EQUATION  YV = GXV  WHERE
C X AND Y ARE MATRICES, V IS THE EIGENVECTOR AND G IS THE
C EIGENVALUE.  THE EIGENVALUES ARE DETERMINED AND PASSED
C BACK TO THE CALLING PROGRAM IN WRODD, WIODD, WREVEN AND
C WIEVEN.
C
      A = ALPHA
      TH = THETA
      REY = REYNO
C
C SET UP MATRIX X FOR ODD EIGENVECTORS.
C
      CALL MSET(CDM,N,MDIM,1,CHM2E1)
C
C INVERT MATRIX X.
C
      CALL CDMTIN(N,CDM,MDIM,DETERM)
C
C SET UP MATRIX Y IN BAND STORAGE MODE FOR ODD EIGENVECTORS.
C
      CALL BMSET(BM,N,MDIM,1,CHM1E1)
C
C MULTIPLY MATRIX Y BY THE INVERSE OF MATRIX X TO CONVERT
C TO THE STANDARD EIGENVALUE PROBLEM WHICH HAS THE FORM
C (Z-G)V = 0   WHERE  Z = (Y)(INVERSE(X)).
C
      CALL MULDBM(CDM,BM,N,5,MDIM,WV)
C
C SPLIT MATRIX INTO REAL AND IMAGINARY PARTS AND CALL
C THE SUBROUTINES TO FIND THE EIGENVALUES.
C
      CALL DSPLIT(N,MDIM,CDM,CDM,DM)
      CALL EHESSC(CDM,DM,1,N,N,MDIM,IVEC)
      CALL ELRH1C(CDM,DM,1,N,N,MDIM,WRODD,WIODD,INERR,IER)
      IF(INERR.NE.0) WRITE(6,9000) INERR,IER
 9000 FORMAT('0ERROR NUMBER',I7,'  ON EIGENVALUE',I7,///)
C
C REPEAT THE SOLUTION FOR EIGENVALUES FOR THE EVEN
C EIGENVECTORS
C
      CALL MSET(CDM,N,MDIM,2,CHM2E1)
      CALL CDMTIN(N,CDM,MDIM,DETERM)
      CALL BMSET(BM,N,MDIM,2,CHM1E1)
      CALL MULDBM(CDM,BM,N,5,MDIM,WV)
      CALL DSPLIT(N,MDIM,CDM,CDM,DM)
      CALL EHESSC(CDM,DM,1,N,N,MDIM,IVEC)
      CALL ELRH1C(CDM,DM,1,N,N,MDIM,WREVEN,WIEVEN,INERR,IER)
      IF(INERR.NE.0) WRITE(6,9000) INERR,IER
      RETURN
      END
```

```
C.......................SUBROUTINE MULDBM...................
C
C
C     PURPOSE
C
C        MULDBM PERFORMS THE MATRIX MULTIPLICATION BETWEEN A
C        SQUARE MATRIX X AND A BANDED MATRIX XB WHICH HAS
C        BEEN SET UP BY SUBROUTINE BMSET.  THE RESULT IS
C        PLACED BACK IN X.  THAT IS...
C                      X = (X)(XB)
C
C     USAGE
C
C        CALL MULDBM(X,XB,N,NB,MDIM,TEMPV)
C
C     DESCRIPTION OF PARAMETERS
C
C        THE FOLLOWING MUST BE SET BY THE CALLING PROGRAM.
C          N,MDIM,NB,X,XB,TEMPV
C
C        N - THE SIZE OF THE MATRICES.
C
C        MDIM - THE DIMENSIONS OF THE MATRICES IN THE CALLING
C          PROGRAM.
C
C        NB - THE NUMBER OF BANDS IN THE BANDED MATRIX, XB.
C
C        X - THE SQUARE N BY N MATRIX.  DIMENSIONED
C          (MDIM,MDIM) IN THE CALLING PROGRAM  (COMPLEX*16)
C
C        XB - THE N BY N MATRIX WHICH IS STORED IN BANDED
C          FORM.  DIMENSIONED (NB,MDIM) IN CALLING PROGRAM.
C          (COMPLEX*16)
C
C        TEMPV - A VECTOR WORKSPACE WHICH MUST BE PROVIDED BY
C          THE CALLING PROGRAM.  DIMENSIONED AT LEAST (N).
C          (COMPLEX*16)
C
C        THE FOLLOWING IS OUTPUT BY MULDBM.
C
C        X - SET TO THE PRODUCT OF X AND XB (MDIM,MDIM)
C          (COMPLEX*16)
C
C     REQUIRED ROUTINES
C
C        NONE
C
C
C...........................................................
C
      SUBROUTINE MULDBM(X,XB,N,NB,MDIM,TEMPV)
      COMPLEX*16 X(MDIM,MDIM),XB(NB,MDIM),TEMPV(MDIM),TEMP
      NBHM = (NB-1)/2
      NBHP = (NB+1)/2
C
C LOOP OVER INDEX I
C
      DO 100 I=1,N
C
C STORE COLUMN I OF MATRIX X TEMPORARILY
C
      DO 10 J=1,N
   10 TEMPV(J) = X(I,J)
C
C FIND PRODUCTS FOR FIRST NBHM SPECIAL CASES, THAT IS WHERE
C WHERE THE BANDED MATRIX DOES NOT HAVE ITS FULL WIDTH
C
      DO 22 J=1,NBHM
      TEMP = (0D0,0D0)
      JJ = NBHM + J
      DO 21 K=1,JJ
      JJJ = JJ-K+1
```

```
   21 TEMP = TEMP+TEMPV(K)*XB(JJJ,K)
   22 X(I,J) = TEMP
C
C COMPUTE PRODUCTS FOR "REGULAR" COMBINATIONS OF ROWS AND
C COLUMNS, THAT IS, THOSE THAT ARE NOT TRUNCATED
C AT THE BEGINNING OR END BY THE BOUNDARIES
C
      JF = N-NBHM
      DO 32 J=NBHP,JF
      TEMP = (0D0,0D0)
      DO 31 K=1,NB
      JJJ = NB-K+1
   31 TEMP = TEMP+TEMPV(J-NBHP+K)*XB(JJJ,J-NBHP+K)
   32 X(I,J) = TEMP
C
C FIND PRODUCTS FOR LAST NBHM SPECIAL CASES.
C
      DO 42 J=1,NBHM
      TEMP = (0D0,0D0)
      JJ = NB-J
      DO 41 K=1,JJ
   41 TEMP = TEMP+TEMPV(N-JJ+K)*XB(NB-K+1,N-JJ+K)
   42 X(I,N-NBHM+J) = TEMP
C
  100 CONTINUE
C
      RETURN
      END
```

```
C.......................SUBROUTINE MSET.........................
C
C
C     PURPOSE
C
C       MSET SETS UP A MATRIX FOR THE FINITE DIFFERENCE
C       PROBLEM OF POISEULLE FLOW WITH THE PROPER BOUNDARY
C       CONDITIONS FOR THE VELOCITY VECTOR POTENTIAL FOR
C       VISCOUS FLOW.
C
C     USAGE
C
C       CALL MSET(X,N,MDIM,MODE,CFMAT)
C
C     DESCRIPTION OF PARAMETERS
C
C       THE FOLLOWING MUST BE SET BY THE CALLING PROGRAM
C         N,MDIM,MODE,CFMAT
C
C       N - THE SIZE OF THE MATRIX.  IF MODE=0, N IS EQUAL
C         TO THE NUMBER OF POINTS OF THE FINITE DIFFERENCE
C         MESH ACROSS THE CHANNEL NOT INCLUDING THE TWO
C         BOUNDARIES.  IF MODE=1 OR MODE=2, N IS EQUAL TO
C         ONE-HALF OF THE NUMBER OF INNER POINTS ACROSS THE
C         FULL CHANNEL.  IN THIS CASE, THE CHANNEL MUST BE
C         DIVIDED INTO AN EVEN NUMBER OF POINTS SO THAT N
C         WILL BE AN INTEGER.
C
C       MDIM - THE COLUMN DIMENSION OF THE MATRIX X.  MDIM
C         MUST BE .GE. N.
C
C       MODE - IF MODE=0, THE MATRIX IS SET UP FOR THE FULL
C         CHANNEL.  IF MODE=1 OR MODE=2, THE MATRIX IS SET
C         UP FOR THE HALF CHANNEL AND THE BOUNDARY
C         CONDITIONS ARE SET SUCH THAT THE ODD OR EVEN
C         EIGENFUNCTIONS, RESPECTIVELY, ARE SOLVED FOR.
C
C       CFMAT - THE NAME OF A FUNCTION SUBPROGRAM WITH TWO
C         PARAMETERS, K AND Y, INDICATING WHICH TERM OF THE
C         FINITE DIFFERENCING IS DESIRED, AND THE POSITION
C         RELATIVE TO THE CENTER OF THE CHANNEL.  MUST BE
C         DECLARED EXTERNAL IN THE CALLING PROGRAM.
C         (COMPLEX*16)
C
C       THE FOLLOWING IS OUTPUT BY MSET
C
C       X - THE N BY N MATRIX INTO WHICH THE COEFFICIENTS
C         OF THE FINITE DIFFERENCING ARE PUT.  MUST BE
C         DIMENSIONED (MDIM,MDIM) IN THE CALLING PROGRAM.
C         (COMPLEX*16)
C
C     NOTES...
C
C       THE EIGENVALUES AND VECTORS OBTAINED BY USING MSET
C         TWICE, WITH MODE=1 AND MODE=2, ARE THE
C         SAME AS USING MSET ONCE WITH MODE=0, BUT WITH
C         N TWICE AS LARGE.
C
C     OTHER ROUTINES NEEDED
C
C       NONE - EXCEPT THE FUNCTION SUBPROGRAM NAME PASSED IN
C       THE PARAMETER 'CFMAT'.
C
C...............................................................
C
      SUBROUTINE MSET(X,N,MDIM,MODE,CFMAT)
      REAL*8 REY,Y,DELY,DFLOAT
      COMPLEX*16 A,G
      REAL*8 TH
      COMMON / COEFNT / A,TH,G,REY,DELY
      COMPLEX*16  CFMAT
      COMPLEX*16 X(MDIM,MDIM)
```

```
C
C COMPUTE GRID SIZE FOR FINITE DIFFERENCE MESH ACROSS
C HALF CHANNEL OR FULL CHANNEL
C
      DELY = 2D0/DFLOAT(N+1)
      IF(MODE.EQ.1.OR.MODE.EQ.2) DELY = 2D0/DFLOAT(2*N+1)
C
C CHECK IF MATRIX DIMENSIONED LARGE ENOUGH
C
      IF(N.LE.MDIM) GO TO 1
      WRITE(6,9000)
 9000 FORMAT('0* * * ERROR - ARRAYS NOT DIMENSIONED LARGE',
     *       ' ENOUGH * * *')
      STOP
C
C ZERO ENTIRE MATRIX
C
    1 DO 10 I=1,N
      DO 10 J=1,N
   10 X(I,J) = (0D0,0D0)
C
C DO SPECIAL CASE AT DISTANCE DELY FROM CHANNEL WALL
C INCLUDING BOUNDARY CONDITIONS
C
      Y = 1D0-DELY
      X(1,1) = CFMAT(3,Y)+CFMAT(1,Y)
      X(1,2) = CFMAT(4,Y)
      X(1,3) = CFMAT(5,Y)
C
C DO SPECIAL CASE AT DISTANCE 2*DELY FROM CHANNEL WALL
C INCLUDING BOUNDARY CONDITIONS
C
      Y = 1D0-2D0*DELY
      X(2,1) = CFMAT(2,Y)
      X(2,2) = CFMAT(3,Y)
      X(2,3) = CFMAT(4,Y)
      X(2,4) = CFMAT(5,Y)
C
C DO ALL REGULAR POINTS IN BETWEEN, THAT IS, THOSE VALUES
C OF Y FOR WHICH ALL 5 FINITE DIFFERENCE GRID POINTS ARE
C INTERIOR TO THE CHANNEL
C
      IL = N-2
      DO 20 I=3,IL
      K = I-3
      Y = 1D0-DELY*DFLOAT(I)
      DO 20 J=1,5
   20 X(I,K+J) = CFMAT(J,Y)
C
C FINALLY DO THE TWO SPECIAL CASES WHICH OCCUR EITHER AT
C THE CENTER OF THE CHANNEL OR AT THE OTHER WALL, DEPENDING
C ON THE VALUE OF MODE.  BOUNDARY CONDITIONS ARE SET UP
C DEPENDING ON MODE
C
      Y = 1D0-DELY*DFLOAT(N-1)
      X(N-1,N-3) = CFMAT(1,Y)
      X(N-1,N-2) = CFMAT(2,Y)
      X(N-1,N-1) = CFMAT(3,Y)
      X(N-1,N) = CFMAT(4,Y)
      IF(MODE.EQ.1) X(N-1,N) = CFMAT(4,Y)-CFMAT(5,Y)
      IF(MODE.EQ.2) X(N-1,N) = CFMAT(4,Y)+CFMAT(5,Y)
C
      Y = 1D0-DELY*DFLOAT(N)
      X(N,N-2) = CFMAT(1,Y)
      X(N,N-1) = CFMAT(2,Y)
      IF(MODE.EQ.1) X(N,N-1) = CFMAT(2,Y)-CFMAT(5,Y)
      IF(MODE.EQ.2) X(N,N-1) = CFMAT(2,Y)+CFMAT(5,Y)
      X(N,N) = CFMAT(3,Y)+CFMAT(5,Y)
      IF(MODE.EQ.1) X(N,N) = CFMAT(3,Y)-CFMAT(4,Y)
      IF(MODE.EQ.2) X(N,N) = CFMAT(3,Y)+CFMAT(4,Y)
C
      RETURN
```

```
      END
C.....................SUBROUTINE BMSET.........................
C
C
C     PURPOSE
C
C       THE PURPOSE OF BMSET IS EXACTLY THAT OF MSET EXCEPT
C       THAT BMSET TAKES ADVANTAGE OF THE BANDED NATURE OF
C       THE FINITE DIFFERENCE MATRICES TO CONSERVE SPACE.
C
C     USAGE
C
C       CALL BMSET(X,N,MDIM,MODE,CFMAT)
C
C     DESCRIPTION OF PARAMETERS
C
C       THE PARAMETERS FOR BMSET ARE THE SAME AS THOSE FOR
C       MSET WITH THE EXCEPTION THAT THE MATRIX X MUST BE
C       DIMENSIONED (5,MDIM) IN THE CALLING PROGRAM.
C
C     NOTE:  THE PROCEDURE IS IDENTICAL TO THAT OF MSET.
C       COMMENTS HAVE THEREFORE NOT BEEN INCLUDED IN BMSET.
C
C
C
C..............................................................
C
      SUBROUTINE BMSET(X,N,MDIM,MODE,CFMAT)
      REAL*8 REY,Y,DELY,DFLOAT
      COMPLEX*16  CFMAT
      COMPLEX*16 A,G
      COMPLEX*16 X(5,MDIM)
      REAL*8 TH
      COMMON / COEFNT / A,TH,G,REY,DELY
      DELY = 2D0/DFLOAT(N+1)
      IF(MODE.EQ.1.OR.MODE.EQ.2) DELY = 2D0/DFLOAT(2*N+1)
      IF(N.LE.MDIM) GO TO 1
      WRITE(6,9000)
 9000 FORMAT('0* * * ERROR - ARRAYS NOT DIMENSIONED LARGE',
     *       ' ENOUGH * * *')
      STOP
    1 DO 10 I=1,5
      DO 10 J=1,MDIM
   10 X(I,J) = (0D0,0D0)
      Y = 1D0-DELY
      X(3,1) = CFMAT(3,Y)+CFMAT(1,Y)
      X(4,1) = CFMAT(4,Y)
      X(5,1) = CFMAT(5,Y)
      Y = 1D0-2D0*DELY
      X(2,2) = CFMAT(2,Y)
      X(3,2) = CFMAT(3,Y)
      X(4,2) = CFMAT(4,Y)
      X(5,2) = CFMAT(5,Y)
      IL = N-2
      DO 20 I=3,IL
      Y = 1D0-DELY*DFLOAT(I)
      DO 20 J=1,5
   20 X(J,I)     = CFMAT(J,Y)
      Y = 1D0-DELY*DFLOAT(N-1)
      X(1,N-1)     = CFMAT(1,Y)
      X(2,N-1)     = CFMAT(2,Y)
      X(3,N-1)     = CFMAT(3,Y)
      X(4,N-1) = CFMAT(4,Y)
      IF(MODE.EQ.1) X(4,N-1) = CFMAT(4,Y)-CFMAT(5,Y)
      IF(MODE.EQ.2) X(4,N-1) = CFMAT(4,Y)+CFMAT(5,Y)
      Y = 1D0-DELY*DFLOAT(N)
      X(1,N)     = CFMAT(1,Y)
      X(2,N)     = CFMAT(2,Y)
      IF(MODE.EQ.1) X(2,N) = CFMAT(2,Y)-CFMAT(5,Y)
      IF(MODE.EQ.2) X(2,N) = CFMAT(2,Y)+CFMAT(5,Y)
      X(3,N) = CFMAT(3,Y)+CFMAT(5,Y)
      IF(MODE.EQ.1) X(3,N) = CFMAT(3,Y)-CFMAT(4,Y)
      IF(MODE.EQ.2) X(3,N) = CFMAT(3,Y)+CFMAT(4,Y)
```

```
      RETURN
      END
C....................SUBROUTINE DSPLIT.........................
C
C
C     PURPOSE
C
C       DSPLIT TAKES A MATRIX OF COMPLEX*16 NUMBERS AND
C       SPLITS IT INTO TWO MATRICES, ONE CONTAINING THE REAL
C       PART OF THE ORIGINAL MATRIX, AND ONE CONTAINING THE
C       IMAGINARY PART.
C
C     USAGE
C
C       CALL DSPLIT(N,MDIM,A,AREAL,AIMAG)
C
C     DESCRIPTION OF PARAMETERS
C
C       N - THE SIZE OF THE MATRIX A, AN N BY N SQUARE
C         MATRIX.
C
C       MDIM - THE COLUMN DIMENSION OF MATRIX A
C
C       A - THE INPUT MATRIX.  MUST BE DIMENSIONED MDIM BY
C         AT LEAST N IN THE CALLING PROGRAM  (COMPLEX*16)
C
C       AREAL,AIMAG - THE OUTPUT MATRICES CONTAINING THE
C         REAL AND IMAGINARY PARTS, RESPECTIVELY, OF
C         MATRIX A.  MUST BE DIMENSIONED (MDIM,MDIM) IN THE
C         CALLING PROGRAM.
C
C     NOTES...
C
C       MATRIX A AND MATRIX AREAL MAY OVERLAP IF THEY ARE
C         DIMENSIONED IN THE CALLING PROGRAM AS FOLLOWS...
C
C               COMPLEX*16 A(MDIM,MDIM)
C               REAL*8 AREAL(MDIM,MDIM),AIMAG(MDIM,MDIM)
C               EQUIVALENCE(A(1,1),AREAL(1,1))
C
C     OTHER ROUTINES NEEDED
C
C       NONE
C
C
C...............................................................
C
      SUBROUTINE DSPLIT(N,MDIM,A,AR,AI)
      REAL*8 A(2,MDIM,MDIM),AR(MDIM,MDIM),AI(MDIM,MDIM)
C
      DO 1 J=1,N
      DO 1 I=1,N
      AR(I,J) = A(1,I,J)
    1 AI(I,J) = A(2,I,J)
C
      RETURN
      END
```

```
C.......................................................................
C
C      SUBROUTINE CDMTIN (CATEGORY F-1)
C
C      PURPOSE
C
C         INVERT A COMPLEX*16 MATRIX
C
C      USAGE
C
C         CALL CDMTIN(N,A,NDIM,DETERM)
C
C      DESCRIPTION OF PARAMETERS
C
C         N      - ORDER OF COMPLEX*16 MATRIX TO BE INVERTED
C                  (INTEGER) MAXIMUM 'N' IS 100
C
C         A      - COMPLEX*16 INPUT MATRIX (DESTROYED).  THE
C                  INVERSE OF 'A' IS RETURNED IN ITS PLACE
C
C         NDIM   - THE SIZE TO WHICH 'A' IS DIMENSIONED
C                  (ROW DIMENSION OF 'A' ACTUALLY APPEARING
C                  IN THE DIMENSION STATMENT OF USER'S
C                  CALLING PROGRAM)
C
C         DETERM - COMPLEX*16 VALUE OF DETERMINANT OF 'A'
C                  RETURNED BY CDMTIN.
C
C      REMARKS
C
C         MATRIX 'A' MUST BE A COMPLEX*8 GENERAL MATRIX
C         IF MATRIX 'A' IS SINGULAR THAT MESSAGE IS PRINTED
C         'N' MUST BE .LE. NDIM
C
C      SUBROUTINES AND FUNCTIONS REQUIRED
C
C         ONLY BUILT-IN FORTRAN FUNCTIONS
C
C      METHOD
C
C         GAUSSIAN ELIMINATION WITH COLUMN PIVOTING IS USED.
C         THE DETERMINANT IS ALSO CALCULATED. A DETERMINANT
C                  OF ZERO INDICATES THAT MATRIX 'A' IS
C                  SINGULAR.
C
C.......................................................................
C
      SUBROUTINE CDMTIN (N,A,NDIM,DETERM)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 A(NDIM,NDIM),PIVOT(100),AMAX,T,SWAP,
     *    DETERM,U
      INTEGER*4 IPIVOT(100),INDEX(100,2)
      REAL*8 TEMP,ALPHA(100)
C
C       INITIALIZATION
C
      DETERM = (1D0,0D0)
      DO 20 J=1,N
      ALPHA(J) = 0D0
      DO 10 I=1,N
   10 ALPHA(J)=ALPHA(J)+A(J,I)*DCONJG(A(J,I))
      ALPHA(J)=DSQRT(ALPHA(J))
   20 IPIVOT(J)=0
      DO 600 I=1,N
C
C       SEARCH FOR PIVOT ELEMENT
C
      AMAX = (0D0,0D0)
      DO 105 J=1,N
      IF (IPIVOT(J)-1) 60,105,60
   60 DO 100 K=1,N
      IF (IPIVOT(K)-1) 80,100,740
```

```
   80 TEMP=AMAX*DCONJG(AMAX)-A(J,K)*DCONJG(A(J,K))
      IF(TEMP) 85, 85, 100
   85 IROW=J
      ICOLUM=K
      AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF(IROW-ICOLUM) 140, 260, 140
  140 DETERM=DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
      SWAP=ALPHA(IROW)
      ALPHA(IROW)=ALPHA(ICOLUM)
      ALPHA(ICOLUM)=SWAP
  260 INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      U=PIVOT(I)
      TEMP=PIVOT(I)*DCONJG(PIVOT(I))
      IF(TEMP) 330, 720, 330
C
C      DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM) = (1D0,0D0)
      DO 350 L=1,N
      U=PIVOT(I)
  350 A(ICOLUM,L)=A(ICOLUM,L)/U
C
C      REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
      IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
      A(L1,ICOLUM) = (0D0,0D0)
      DO 450 L=1,N
      U=A(ICOLUM,L)
  450 A(L1,L)=A(L1,L)-U*T
  550 CONTINUE
  600 CONTINUE
C
C      INTERCHANGE COLUMNS
C
  620 DO 710 I=1,N
      L=N+1-I
      IF(INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
      RETURN
  720 WRITE(6,730)
  730 FORMAT(20H  MATRIX IS SINGULAR)
  740 RETURN
      END
```

```
C.EHESSC.........D.............................................
C
C     FUNCTION                 REDUCTION OF A COMPLEX MATRIX TO
C                                 UPPER HESSENBERG FORM.
C     USAGE                  - CALL EHESSC(AR,AI,K,L,N,IA,ID)
C     PARAMETERS     AR      - INPUT/OUTPUT MATRIX OF DIMENSION
C                                 N BY N.
C                                 ON INPUT CONTAINS THE REAL
C                                 COMPONENTS OF THE REDUCED HESSEN
C                                 BERG FORM IN UPPER TRIANGULAR
C                                 PORTION AND THE DETAILS OF THE
C                                 REDUCTION IN LOWER TRIANGULAR
C                                 PORTION.
C                    AI      - INPUT/OUTPUT N BY N MATRIX
C                                 CONTAINING THE IMAGINARY COUNTER
C                                 PARTS TO AR, ABOVE.
C                    K       - INPUT SCALAR CONTAINING THE ROW
C                                 AND COLUMN INDEX OF THE STARTING
C                                 ELEMENT TO BE REDUCED BY ROW
C                                 SCALING. FOR UNBALANCED
C                                 MATRICES SET  L = N.
C                    L       - INPUT SCALAR CONTAINING THE ROW
C                                 AND COLUMN INDEX OF THE LAST
C                                 ELEMENT TO BE REDUCED BY ROW
C                                 SCALING. FOR UNBALANCED
C                                 MATRICES SET K = 1.
C                    N       - INPUT SCALAR CONTAINING THE ORDER
C                                 OF THE MATRIX TO BE REDUCED.
C                    IA      - INPUT SCALAR CONTAINING ROW
C                                 DIMENSION OF AR AND AI IN THE
C                                 CALLING PROGRAM.
C                    ID      - OUTPUT VECTOR OF LENGTH L CONTAIN
C                                 ING DETAILS OF THE
C                                 TRANSFORMATION.
C     PRECISION              - SINGLE/DOUBLE
C     CODE RESPONSIBILITY    - T.J. AIRD/E.W. CHOU
C     LANGUAGE               - FORTRAN
C...............................................................
C     LATEST REVISION        - FEBRUARY 7, 1973
C
      SUBROUTINE EHESSC (AR,AI,K,L,N,IA,ID)
C
      DIMENSION              AR(IA,1),AI(IA,1),ID(1),T1(2),T2(2)
      DOUBLE PRECISION       AR,AI,XR,XI,YR,YI,T1,T2,ZERO
      COMPLEX*16             X,Y
      EQUIVALENCE        (X,T1(1),XR),(T1(2),XI),(Y,T2(1),YR),
     1                       (T2(2),YI)
      DATA                   ZERO/0.0D0/
      LA=L-1
      KP1=K+1
      IF (LA .LT. KP1) GO TO 45
      DO 40 M=KP1,LA
         I=M
         XR=ZERO
         XI=ZERO
         DO 5 J=M,L
            IF(DABS(AR(J,M-1))+DABS(AI(J,M-1)).LE.
     *         DABS(XR)+DABS(XI))
     1         GO TO 5
            XR=AR(J,M-1)
            XI=AI(J,M-1)
            I=J
    5    CONTINUE
         ID(M)=I
         IF (I .EQ. M) GO TO 20
C                              INTERCHANGE ROWS AND COLUMNS
C                                          ARRAYS AR AND AI
         MM1=M-1
         DO 10 J=MM1,N
            YR=AR(I,J)
            AR(I,J)=AR(M,J)
            AR(M,J)=YR
```

```
            YI=AI(I,J)
            AI(I,J)=AI(M,J)
            AI(M,J)=YI
10       CONTINUE
         DO 15 J=1,L
            YR=AR(J,I)
            AR(J,I)=AR(J,M)
            AR(J,M)=YR
            YI=AI(J,I)
            AI(J,I)=AI(J,M)
            AI(J,M)=YI
15       CONTINUE
C                                           END INTERCHANGE
20       IF (XR .EQ. ZERO .AND. XI .EQ. ZERO) GO TO 40
         MP1=M+1
         DO 35 I=MP1,L
            YR=AR(I,M-1)
            YI=AI(I,M-1)
            IF (YR .EQ. ZERO .AND. YI .EQ. ZERO) GO TO 35
            Y=Y/X
            AR(I,M-1)=YR
            AI(I,M-1)=YI
            DO 25 J=M,N
               AR(I,J)=AR(I,J)-YR*AR(M,J)+YI*AI(M,J)
               AI(I,J)=AI(I,J)-YR*AI(M,J)-YI*AR(M,J)
25          CONTINUE
            DO 30 J=1,L
               AR(J,M)=AR(J,M)+YR*AR(J,I)-YI*AI(J,I)
               AI(J,M)=AI(J,M)+YR*AI(J,I)+YI*AR(J,I)
30          CONTINUE
35       CONTINUE
40    CONTINUE
45    RETURN
      END
```

```
C.ELRH1C........D..............................................
C
C    FUNCTION                 - COMPUTATION OF ALL EIGENVALUES OF
C                                 A COMPLEX UPPER HESSENBERG
C                                 MATRIX.
C    USAGE                    - CALL ELRH1C(HR,HI,K,L,N,IH,WR,WI,
C                                 INFER,IER)
C    PARAMETER     HR         - INPUT MATRIX OF DIMENSION N BY N
C                                 CONTAINING THE REAL COMPONENTS
C                                 OF THE COMPLEX HESSENBERG MATRIX
C                                   HR IS DESTROYED ON OUTPUT.
C                  HI         - INPUT MATRIX OF DIMENSION N BY N
C                                 CONTAINING THE IMAGINARY COUNTER
C                                 PARTS TO HE, ABOVE.  HI IS
C                                 DESTROYED ON OUTPUT.
C                  K          - INPUT SCALAR CONTAINING THE LOWER
C                                 BOUNDARY INDEX FOR THE INPUT
C                                 MATRIX.  FOR UNBALANCED MATRICES
C                                 SET K = 1.
C                  K          - INPUT SCALAR CONTAINING THE UPPER
C                                 BOUNDARY INDEX FOR THE INPUT
C                                 MATRIX.  FOR UNBALANCED MATRICES
C                                 SET L = N.
C                  N          - INPUT SCALAR CONTAINING THE ORDER
C                                 OF THE MATRIX.
C                  IH         - INPUT SCALAR CONTAINING THE ROW
C                                 DIMENSION OF MATRICES HR AND HI,
C                                 IN THE CALLING PROGRAM.
C                  WR         OUTPUT VECTOR OF LENGTH N CONTAIN
C                                 ING COMPONENTS OF THE
C                                 EIGENVALUES.
C                  WI         OUTPUT VECTOR OF LENGTH N CONTAIN
C                                 ING IMAGINARY COMPONENTS OF THE
C                                 EIGENVALUES.
C                  INFER      - OUTPUT SCALAR CONTAINING THE INDEX
C                                 OF THE EIGENVALUE WHICH
C                                 GENERATED THE TERMINAL ERROR.
C                                   N=1 INDICATED THE EIGENVALUE
C                                         RECORDED IN THE OUTPUT
C                                         PARAMETER, INFER, COULD
C                                         NOT BE DETERMINED AFTER 30
C                                         ITERATIONS. IF THE J-TH
C                                         EIGENVALUE COULD NOT BE SO
C                                         DETERMINED, THEN THE EIGEN
C                                         VALUES J+1,J+2,...,N
C                                         SHOULD BE CORRECT.
C    PRECISION                - SINGLE/DOUBLE
C    REQ'D IMSL ROUTINES      - UERTST
C    CODE RESPONSIBILITY      - T.J. AIRD/E.W. CHOU
C    LANGUAGE                 - FORTRAN
C..............................................................
C    LATEST REVISION          - MARCH 22, 1973
C
      SUBROUTINE ELRH1C (HR,HI,K,L,N,IH,WR,WI,INFER,IER)
C
      DIMENSION          HR(IH,1),HI(IH,1),WR(1),WI(1)
      DIMENSION          T1(2),T2(2),T3(2)
      COMPLEX*16         X,Y,Z
      DOUBLE PRECISION HR,HI,WR,WI,EPS,ZR,ZI,T1,T2,T3,RDELP
      DOUBLE PRECISION ZERO,ONE,TWO,SR,SI,XR,XI,YR,RI,TR,TI
      EQUIVALENCE        (X,T1(1),XR),(T1(2),XI),
     1                   (Y,T2(1),YR),(T2(2),YI),
     2                   (Z,T3(1),ZR),(T3(2),ZI)
      DATA               TWO/2.0D0/
      DATA    ZERO,ONE,RDELP/0.D0,1.D0,Z3410000000000000/
      INFER=0
      IER=0
      DO 5 I=1,N
         IF (I .GE. K .AND. I .LE. L) GO TO 5
         WR(I)=HR(I,I)
         WI(I)=HI(I,I)
    5 CONTINUE
```

```
      NN=L
      TR=ZERO
      TI=ZERO
C                              SEARCH FOR NEXT EIGENVALUE
   10 IF (NN .LT. K) GO TO 9005
      ITS=0
      NM1=NN-1
      IF (NN .EQ. K) GO TO 25
C                              LOOK FOR SINGLE SMALL SUB-DIAGONAL
C                                             ELEMENTS
   15 NPL=NN+K
      DO 20 LL=K,NM1
         M=NPL-LL
         MM1=M-1
         IF (DABS(HR(M,MM1)) + DABS(HI(M,MM1)) .LE.
     1    RDELP*(DABS(HR(MM1,MM1)) + DABS(HI(MM1,MM1)) +
     2       DABS(HR(M,M)) + DABS(HI(M,M)))) GO TO 30
   20 CONTINUE
   25 M=K
   30 IF (M .EQ. NN) GO TO 110
      IF (ITS .EQ. 30) GO TO 115
C                                          FORM SHIFT
      IF (ITS .EQ. 10 .OR. ITS .EQ. 20) GO TO 35
      SR=HR(NN,NN)
      SI=HI(NN,NN)
      XR=HR(NM1,NN)*HR(NN,NM1)-HI(NM1,NN)*HI(NN,NM1)
      XI=HR(NM1,NN)*HI(NN,NM1)+HI(NM1,NN)*HR(NN,NM1)
      IF (XR .EQ. ZERO .AND. XI .EQ. ZERO) GO TO 40
      YR=(HR(NM1,NM1)-SR)/TWO
      YI=(HI(NM1,NM1)-SI)/TWO
      Z=CDSQRT(DCMPLX(YR**2-YI**2+XR,TWO*YR*YI+XI))
      IF (YR*ZR+YI*ZI .LT. ZERO) Z=-Z
      X=X/(Y+Z)
      SR=SR-XR
      SI=SI-XI
      GO TO 40
   35 SR=DABS(HR(NN,NM1))+DABS(HR(NM1,NN-2))
      SI=DABS(HI(NN,NM1))+DABS(HI(NM1,NN-2))
   40 DO 45 I=K,NN
         HR(I,I)=HR(I,I)-SR
         HI(I,I)=HI(I,I)-SI
   45 CONTINUE
      TR=TR+SR
      TI=TI+SI
      ITS=ITS+1
C                              LOOK FOR TWO CONSECUTIVE SMALL
C                                             SUB-DIAGONAL ELEMENTS
      XR=DABS(HR(NM1,NM1))+DABS(HI(NM1,NM1))
      YR=DABS(HR(NN,NM1))+DABS(HI(NN,NM1))
      ZR=DABS(HR(NN,NN))+DABS(HI(NN,NN))
      NMJ=NM1-M
      IF (NMJ .EQ. 0) GO TO 55
C                              FOR MM=NN-1 STEP -1 UNTIL M+1 DO
      DO 50 J=1,NMJ
         MM=NN-J
         M1=MM-1
         YI=YR
         YR=DABS(HR(MM,M1))+DABS(HI(MM,M1))
         XI=ZR
         ZR=XR
         XR=DABS(HR(M1,M1))+DABS(HI(M1,M1))
         IF (YR.LE.RDELP*ZR/YI*(ZR+XR+XI)) GO TO 60
   50 CONTINUE
   55 MM=M
C                                          TRIANGULAR DECOMPOSITION
   60 MP1=MM+1
      DO 85 I=MP1,NN
         IM1=I-1
         XR=HR(IM1,IM1)
         XI=HI(IM1,IM1)
         YR=HR(I,IM1)
         YI=HI(I,IM1)
```

```
         IF(DABS(XR)+DABS(XI).GE.DABS(YR)+DABS(YI)) GO TO 70
C                       INTERCHANGE ROWS OF HR AND HI
         DO 65 J=IM1,NN
            ZR=HR(IM1,J)
            HR(IM1,J)=HR(I,J)
            HR(I,J)=ZR
            ZI=HI(IM1,J)
            HI(IM1,J)=HI(I,J)
            HI(I,J)=ZI
   65    CONTINUE
         Z=X/Y
         WR(I)=ONE
         GO TO 75
   70    Z=Y/X
         WR(I)=-ONE
   75    HR(I,IM1)=ZR
         HI(I,IM1)=ZI
         DO 80 J=I,NN
            HR(I,J)=HR(I,J)-ZR*HR(IM1,J)+ZI*HI(IM1,J)
            HI(I,J)=HI(I,J)-ZR*HI(IM1,J)-ZI*HR(IM1,J)
   80    CONTINUE
   85 CONTINUE
C                                   COMPOSITION
      DO 105 J=MP1,NN
         JM1=J-1
         XR=HR(J,JM1)
         XI=HI(J,JM1)
         HR(J,JM1)=ZERO
         HI(J,JM1)=ZERO
C                       INTERCHANGE COLUMNS OF HR AND HI IF
C                                      NECESSARY
         IF (WR(J) .LE. ZERO) GO TO 95
         DO 90 I=M,J
            ZR=HR(I,JM1)
            HR(I,JM1)=HR(I,J)
            HR(I,J)=ZR
            ZI=HI(I,JM1)
            HI(I,JM1)=HI(I,J)
            HI(I,J)=ZI
   90    CONTINUE
   95    DO 100 I=M,J
            HR(I,JM1)=HR(I,JM1)+XR*HR(I,J)-XI*HI(I,J)
            HI(I,JM1)=HI(I,JM1)+XR*HI(I,J)+XI*HR(I,J)
  100    CONTINUE
  105 CONTINUE
      GO TO 15
C                                   A ROOT FOUND
  110 WR(NN)=HR(NN,NN)+TR
      WI(NN)=HI(NN,NN)+TI
      NN=NM1
      GO TO 10
C                       SET ERROR-NO CONVERGENCE TO AN
C                           EIGENVALUE AFTER 30 ITERATIONS
  115 INFER=NN
      IER=129
 9000 CONTINUE
      CALL UERTST (IER,6HELRH1C)
 9005 RETURN
      END
```

```
C      SUBROUTINE UERTST (IER,NAME)
C
C-UERTST----------------LIBRARY--------------------------------
C
C   FUNCTION               - ERROR MESSAGE GENERATION
C   USAGE                  - CALL UERTST(IER,NAME)
C   PARAMETERS   IER       - ERROR PARAMETER. TYPE + N  WHERE
C                              TYPE= 128 IMPLIES TERMINAL ERROR
C                                     64 IMPLIES WARNING WITH FIX
C                                     32 IMPLIES WARNING
C                              N    = ERROR CODE RELEVANT TO
C                                     CALLING ROUTINE.
C                NAME      - INPUT VECTOR CONTAINING THE NAME
C                              OF THE CALLING ROUTINE AS A SIX
C                              CHARACTER LITERAL STRING.
C   LANGUAGE               - FORTRAN
C--------------------------------------------------------------
C   LATEST REVISION        - JANUARY 18, 1974
C
      SUBROUTINE UERTST(IER,NAME)
C
      DIMENSION             ITYP(5,4),IBIT(4)
      INTEGER*2             NAME(3)
      INTEGER               WARN,WARF,TERM,PRINTR
      EQUIVALENCE (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),
     *            TERM)
      DATA  ITYP  /'WARN','ING ','    ','    ','    ',
     *             'WARN','ING(','WITH',' FIX',')   ',
     *             'TERM','INAL','    ','    ','    ',
     *      IBIT      / 32,64,128,01
      DATA      PRINTR     / 6/
      IER2=IER
      IF (IER2 .GE. WARN) GO TO 5
C                                   NON-DEFINED
      IER1=4
      GO TO 20
    5 IF (IER2 .LT. TERM) GO TO 10
C                                   TERMINAL
      IER1=3
      GO TO 20
   10 IF (IER2 .LT. WARF) GO TO 15
C                                   WARNING(WITH FIX)
      IER1=2
      GO TO 20
C                                   WARNING
   15 IER1=1
C                                   EXTRACT 'N'
   20 IER2=IER2-IBIT(IER1)
C                                   PRINT ERROR MESSAGE
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,5),NAME,IER2,IER
   25 FORMAT(' *** I M S L(UERTST) ***  ',5A4,4X,3A2,4X,I2,
     *   ' (IER = ',I3,')')
      RETURN
      END
```

```
C.................FUNCTION CHM1E1 AND CHM2E1....................
C...............................................................
C                     (CARTESIAN COORDINATES)
C
C
C      PURPOSE
C
C         CHM1E1 AND CHM2E1 RETURN THE VALUES (COMPLEX*16) OF
C         THE COEFFICIENTS FOR THE MATRICES IN THE FINITE
C         DIFFERENCE FORM OF THE LINEARIZED NAVIER-STOKES
C         EQUATION FOR POISEUILLE FLOW.  BOTH FUNCTIONS RESULT
C         FROM THE LINEAR COMBINATION OF EQUATION 1 AND
C         EQUATION 3 TO ELIMINATE THE VELOCITY VECTOR
C         POTENTIAL COMPONENT G AND ARBITRARILY SETTING THE
C         COMPONENT F TO ZERO.  SO, THEY ARE THE COEFFICIENTS
C         FOR THE VECTOR POTENTIAL COMPONENT H.  CHM2E1
C         RETURNS THE TERMS WHICH ARE COEFFICIENTS OF THE
C         EIGENVALUE, GAMMA, AND CHM1E1 RETURNS THE REMAINING
C         TERMS.
C
C      USAGE
C
C         X1 = CHM1E1(K,Y)
C         X2 = CHM2E1(K,Y)
C
C           (CHM1E1 AND CHM2E1 MUST BE DECLARED COMPLEX*16 IN
C           CALLING PROGRAM)
C
C      DESCRIPTION OF PARAMETERS
C
C         THE FOLLOWING PARAMETERS MUST BE SET BY THE CALLING
C           PROGRAM
C
C         K - INDICATES THE POINT ON THE FINITE DIFFERENCE
C           MESH RELATIVE TO THE CENTRAL POINT IN THE CENTRAL
C           DIFFERENCING SCHEME.  IF THE DIFFERENCE IS BEING
C           FORMED ABOUT THE N-TH POINT THEN K=1 REFERS TO
C           THE POINT N-2, K=2 REFERS TO THE POINT N-1,
C           K=3 REFERS TO N, K=4 REFERS TO N+1, AND K=5
C           REFERS TO N+2.
C         K - INDICATES WHICH POINT ON THE FINITE DIFFERENCE
C           MESH IS REFERRED TO THE CENTRAL POINT.  IF THE
C           DIFFERENCE IS BEING FORMED ABOUT THE N-TH POINT
C           THEN K=1 REFERS TO THE POINT N-2, K=2 REFERS TO
C           THE POINT N-1, K=3 REFERS TO N, K=4 REFERS TC N+1,
C           AND K=5 REFERS TO N+2.
C
C         Y - THE VALUE OF THE POSITION RELATIVE TO THE CENTER
C           OF THE CHANNEL.  THE TWO BOUNDARIES ARE AT Y=+1
C           AND Y=-1.
C
C      OTHER ROUTINES NEEDED
C
C         NONE
C
C
C...............................................................
C
       FUNCTION CHM1E1(K,Y)
       IMPLICIT COMPLEX*16 (A-H,O-Z)
       COMMON / COEFNT / A,TH,G,REY,DEL
       REAL*8 REY,Y,DEL
       REAL*8 TH,DUR
C
C THE FOLLOWING FUNCTIONS (M1) EVALUATE THE COEFFICIENTS
C OF THE DERIVATIVES OF H FOR ALL TERMS EXCEPT THOSE
C CONTAINING GAMMA.
C
       CH4M1(Y) = A*EI/REY
       CH2M1(Y) = -1.5D0*A**2*EI2*(1D0-Y**2)+2D0*AEI*(A**2)/
     *       REY
       CH0M1(Y) = -AEI*((A**2)*(1.5D0*AEI*(1D0-Y**2)-(A**2)
```

```
     *    /REY)+3D0*AEI)
C
C THE REMAINING FUNCTIONS (M2) EVALUATE THE COEFFICIENTS
C OF THE DERIVATIVES OF H WHICH ARE ALSO COEFFICIENTS
C OF THE EIGENVALUE GAMMA.
C
      CH2M2(Y) = AEI
      CHOM2(Y) = AEI * A**2
      DUR = 0.0
      DU = DCMPLX(DUR,TH)
      EI = CDEXP(DU)
      AEI = A*EI
      EI2 = CDEXP(2*DU)
C
C SET UP THE FINITE DIFFERENCE VALUES FOR INDEX K FOR M1
C
      GO TO (1,2,3,2,1),K
    1 CHM1E1 = CH4M1(Y)/DEL**4
      GO TO 100
    2 CHM1E1 = -4D0*CH4M1(Y)/DEL**4+CH2M1(Y)/DEL**2
      GO TO 100
    3 CHM1E1 = 6D0*CH4M1(Y)/DEL**4-2D0*CH2M1(Y)/DEL**2
     *    +CHOM1(Y)
  100 RETURN
C
C SET UP THE FINITE DIFFERENCE VALUES FOR INDEX K FOR M2
C
      ENTRY CHM2E1(K,Y)
      DUR = 0.0
      DU = DCMPLX(DUR,TH)
      EI = CDEXP(DU)
      AEI = A*EI
      EI2 = CDEXP(2.0*DU)
      GO TO (11,12,13,12,11),K
   11 CHM2E1 = (0D0,0D0)
      GO TO 200
   12 CHM2E1 = CH2M2(Y)/DEL**2
      GO TO 200
   13 CHM2E1 = -2D0*CH2M2(Y)/DEL**2+CHOM2(Y)
  200 RETURN
      END
```

LIST OF REFERENCES

1. Harrison, W. F., On the Stability of Poiseuille Flow, M.S. Thesis, Naval Postgraduate School, 1975.

2. Salwen, H., and Grosch, C. E., "The Stability of Poiseuille Flow in a Pipe of Circular Cross-section," Journal of Fluid Mechanics, v. 54, part 1, p. 93, 6 March 1972.

3. Garg, V. W., and Rouleau, W. T., "Linear Spatial Stability of Pipe Poiseuille Flow," Journal of Fluid Mechanics, v. 54, part 1, p. 113, 25 November 1971.

4. Schlichting, H., Boundary Layer Theory, McGraw-Hill 1968, p. 516.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 57 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Professor T. H. Gawain, Code 57Gn Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	LCDR Lewis R. Newby, USN 1026 Victory Drive St. Louis, Missouri 63125	1